MARINE SPILL SIMULATION SOFTWARE SET

by J. J. A. van Huijstee

Hydraulic Engineering Group
Department of Civil Engineering
Delft University of Technology
Delft, The Netherlands

February 1985

Marine Spill Simulation Software Set

by J. J. A. van Huijstee

february 1985

Student Thesis

Coastal Engineering Group

Department of Civil Engineering

Delft University of Technology

Delft, The Netherlands

This work has been carried out under the general supervision of Prof. dr. ir. E. W. Bijker, professor of Coastal Engineering, and the detailed supervision of W. W. Massie, MSc., P.E., Senior Scientific officer of the Coastal Engineering Group.

Contents

Appendix E: MS4-1: Data accumulation and processing (seperate volume)
- List of all files
- Source programs (first part)

Appendix F: MS4-1: Data accumulation and processing (seperate volume)
- Source programs (second part)
- System files
- List of data files directory (EXTRA)
- Data files directory (EXTRA)

Appendix G: MS4-2: Actual simulation (seperate volume)
- List of all files
- Source programs and system files

Appendix H: MS4-3: Output presentation (seperate volume)
- List of all files
- Source program and system files

Appendix I: Program testing (seperate volume)
Simulation 1: 100 minutes

Appendix J: Program testing (seperate volume)
Simulation 2: 100 hours

Appendix K: MS4-1: Data accumulation and processing (seperate volume)
- BASIC program to plot a map
- Example of plotted map
- Data file

Abstract

The Marine Spill Simulation Software Set is a tool which allows the user to simulate the fate of an oil spill at sea. The software set and the simulation structure have been designed in such a way that the friendly interaction between user and computer has become a most salient feature.

Summary

The Marine Spill Simulation Software Set is based on physical and information theoretical components.

The physical component of the simulation model consists of a system of differential and algebraic equations that describe processes which influence the motions and characteristics of oil at sea.

This report deals especially with the information theoretical components of the simulation model.

At the beginning of this report the motivations in choosing a microcomputer for this application - instead of a mainframe - are explained. Thereupon attention is transferred from hardware to software. The reasons for selecting Fortran 77 as programming language are stated and the user-friendly elements of the model are discussed and illustrated with a few examples. The simulation software structure clearly shows that the model is divided in three major modules namely a data accumulation and processing module, an actual simulation module and an output module. All input data are summed up and the way these data are handled is discussed. The present-day required processing of input data and future input possibilities bring the survey of the data accumulation and processing module to a close. Efficient integration methods for the simulation processes are selected to improve run speed. The handling of output data and the recommended future output presentation are discussed next. Finally, simulation runs are done with test data to check is the model functions correctly. Furthermore run time, accuracy, efficiency and stability are determined and assessed.

To conclude this summary the reader is kindly recommended to try out the Marine Spill Simulation Software Set.

1. Introduction

1.1. A retrospective view: SMOSS

In February 1980 Herman D. Kuipers presented the first part of his engineers thesis in which processes that influence the motions and characteristics of oil at sea were studied.

A year later the publication of SMOSS completed his thesis. SMOSS, an acronym of the full title of the second part of his thesis - A Simulation Model for Oil Slicks at Sea -, is mostly used to reference the actual computer program which is able to predict transport, spreading and some aging processes which are significant for oil combatting actions.

SMOSS includes the following items in simulating the fate of an initial oil spill or oil leak:

- spread due to five forces : a. Gravitational force
  b. Net surface tension
  c. Inertia force
  d. Internal viscous force
  e. Viscous force

- slick transport caused by : a. Waves
  b. Winds
  c. Tides
  d. Currents

- the following aging processes: a. Evaporation
  b. Emulsification
  c. Dispersion
  d. Dissolution
  e. Direct sea-air exchange

- changes in : a. flashpoint
  b. oil density
  c. oil viscosity
  d. net surface tension

- time at which: a. the oil will sink
  b. the oil spill will break up

For more detailed information on these items reference is made to appendix A or the work of Herman D. Kuipers (see References).

Because the processes in the computer program are described by differential and algebraic equations, the Continuous System Modeling Program (CSMP III) was chosen because of its ability to solve a system of differential equations.

On the basis of positive experience it was decided to continue the "SMOSS-project". This thesis is the first impulse to improve and enlarge the marine spill simulation model.

1.2. A new approach: MS4

The Marine Spill Simulation Software Set, which during the course of this report will be referred to as MS4, is made up - already indicated by the name - of a software set consisting of many computer programs, data files, etcetera.

MS4 has been developed on a Digital PROFESSIONAL 350 microcomputer using FORTRAN 77 as a primary programming language. The choice of hard- and software was primarily made on the basis of flexibility and efficiency.

The modular and clearly structured software set is closely bound to the most salient feature of MS4 which is the friendly interaction between user and computer.

In taking a closer look at the above mentioned software structure, MS4 can be seen divided in three main segments. Namely the handling and processing of input data, the main program in which the actual simulation takes place and last but not least the presentation of output data.

The strict seperation of these three segments has resulted in increased input/output flexibility and decreased simulation run time. The additional use of screen displayed data forms and software attributes commonly referred to as menus and help text displays throughout the whole simulation has made MS4 accesible to those unacquainted with programming.

The Marine Spill Simulation Software Set objectives are:

- Convert the SMOSS programming language from CSMP III to FORTRAN 77

* See appendices E, F, G and H
  See paragraph 2.2.

- Speed up computer run time

* See paragraphs 5.1. and 7.5.

- Improve presentation simulation:
  a. Installed applications
  b. Single choice menus
  c. Help texts

* See paragraph 2.3.

- Build a modular structured simulation model

* See chapter 3.
  See appendix C.

- Improve data accumulation:
  a. Class all input data in seperate categories
  b. Electronic forms
  c. Warning messages
  d. Use of Gregorian calendar for input of time dependent data
  e. Use of data bases for regular returning input data
  f. Begin of a visual display on the screen of the spill on the map

* a/d/e/f See chapter 4.
  b/c See paragraph 2.3.
  f See appendix K

- Improve processing: a. A different interpolation method
  b. Uncouple slick velocity computation from the actual simulation
  c. Make the processing-module self-contained to make background processing (spawning) possible

* See chapter 4.

- Improve actual simulation: a. Select another integration routine
  b. Install a time clock

* See chapter 5.

- Improve output presentation: a. All output with a date and time label
  b. File prints possible at each stage of the simulation

* See chapter 6.

- Check the new simulation model

* See paragraphs 7.1., 7.2. and 7.3.

- Determine accuracy, efficiency and stability model

* See paragraphs 7.4., 7.6. and 7.7.

- List conclusions and recommendations

* See chapter 8.

2. From hardware to software

2.1. Why choose a microcomputer?

The decision to transfer the simulation model from a mainframe to a microcomputer was largely made on the basis of the following considerations:

- The Continuous System Modeling Program (CSMP III), as used in SMOSS, is a software package that is usually only implemented on large IBM machines. By removing this limiting condition, the simulation model comes within the reach of a larger group of potential users.

- Based on the experience with SMOSS - a memory requirement of several hundreds of kilobytes and the use of REAL*8 numbers only - an absolute condition was that these variables would also be available and that - as a minimum - a 16 bit processor with overlay possibilities be available to carry out the computations.

- In general the microcomputer is far more user friendly than a mainframe.

- Growth of microcomputer potentialities effectuates a growing interest for scientific and engineering applications on microcomputers. A tendency to bear in mind.

2.2. The computer language: FORTRAN 77

FORTRAN, which is an acronym of FORmula TRANslator, was the earliest non-machine-specific high-level language to be specified, and it is still the most widely used language in scientific and engineering applications today.

MS4 compromises calculations of great magnitude and complexity. For a large problem like this, the critical issue is the way data are handled: how they are input to the system, and how they are stored inside the available memory and processing space. To ensure that data are stored efficiently, it is desirable to distinguish between different types of data. FORTRAN, by contrast with BASIC, has very precise formatting control procedures covering the ways in which input and output data are organized. Furthermore FORTRAN results - via compilation - in a faster running end program than BASIC that generally uses an interpreter. However, compiled BASIC might well be just about as fast.

FORTRAN has been designed to allow for greater flexibility than BASIC in the definition of functions and subroutines.

FORTRAN is the best-defined and standardized high-level language. Its standards have not only been efficiently developed; they are also widely observed - a fact that is not true of many other so-called standardized high-level languages. Therefore adapting a FORTRAN program to run on a different computer is a comparetively trivial business. This vital feature of FORTRAN emphasizes its continuing importance, particularly in the professional programming environment.

2.3. User-friendly interaction with the computer

A complete description of the MS4 interaction between user and computer, which can excellently be used as manual, can be found in appendix B. The purpose of this paragraph is to give insight in the user friendly components in MS4.

The reasons to strive for a friendly interaction between the user and computer are various:

- it makes the model more enjoyable to work with
- fewer mistakes are to be expected
- no programming experience is required

The MS4 user friendly components are:

- installed applications
- single choice menus (see figure 2.1.)
- help texts
- forms (see figure 2.2.)
- warning messages (see figure 2.2.)

```
Marine Spill Simulation Software Set

Choose from the following options:
        →
          Prepare input for new simulation
          Modify input previous simulation
          Exit

Make a choice and press the DO key:
```

Figure 2.1. A single choice menu.

Installed applications

MS4 has been installed as three (MS4-1, MS4-2 and MS4-3) seperate applications. In practice this means that "run" or similar commands have become superfluous. In following this procedure one restriction remains namely that prior to running the installed application the corresponding program directory must have been selected.
The alternative procedure in starting the simulation is by selecting the Tool kit option from the main menu and giving the right commands; see appendix B.5..

Single choice menus

These menus function as nodal points of the application. By selecting from the options, the operator decides which action the computer is to take.

Help texts

By pressing the HELP key on the keyboard a help text will appear on the screen giving information with regard to problems the user is likely to encounter in that stage of the simulation.

Forms

Data accumulation takes place by filling in forms that appear on the screen. The sequence of input is dictated by each form.

Warning messages

In case a severe mistake is made in filling in a form, a single line warning message will appear at the bottom of the screen. The user can then correct the error.

```
Input of time data

Interpolation

Enter date commencement interpolation: Year : 1984
                                       Month: 02
                                       Day  : __

Illegal input. February has only 29 days.
```

Figure 2.2. Form with warning message.

3. The simulation software structure

The Marine Spill Simulation Software Set has been developed on a Digital Professional 350. See appendix D.

Fortran 77 has been used as programming language. See paragraph 2.2..

The Marine Spill Simulation Software Set structure consists of three major and four minor modules. Each module is stored in a seperate directory.

The three major modules carry out the simulation:

Directory	Module	Function
MS4	MS4-1	Data accumulation and processing. See chapter 4. See appendices E and F.
MS5	MS4-2	The actual simulation. See chapter 5. See appendix G.
MS6	MS4-3	Output presentation. See chapter 6. See appendix H.

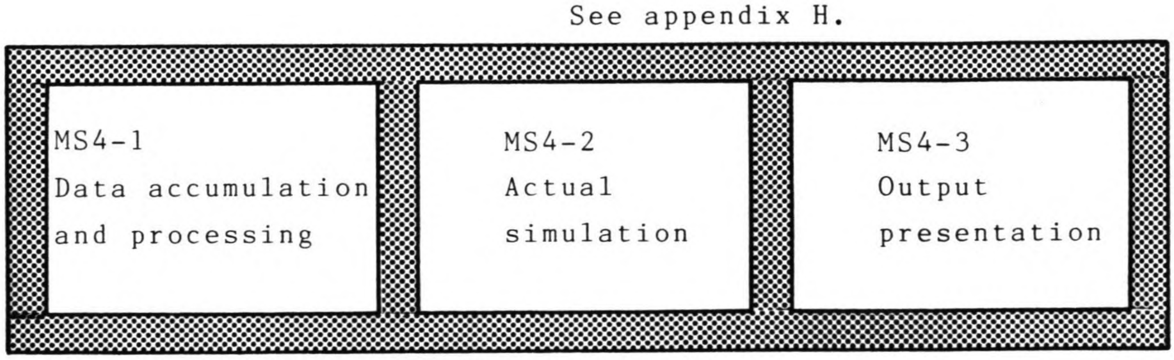


Figure 3.1. Three major modules that carry out the simulation.

Interaction user-computer major modules: Appendix B.
Program file structure major modules : Appendix C.

The four minor modules provide storage for all kinds of data:

Directory	Function
SDF	Storage of input data coming from the data accumulation phase (MS4-1) that require processing (MS4-1) before they can be used as input to the actual simulation (MS4-2).
DF	Storage of input data coming from the data accumulation phase (MS4-1) that do not require processing. Storage of data that result from processing (MS4-1) and can be used as input to the actual simulation.
EXTRA	Storage of unalterable data required by data accumulation programs (MS4-1).
OUT	Storage of output data.

A more detailed illustration of the simulation software structure is given in appendix C.

4. Data accumulation and processing

4.1. Input data

From the point of view of simulation flexibility, clarity of arrangement and a modular structure the first requisite is to point out all the possible input categories for the marine spill simulation.

a. Time
b. Place
c. Chemical technology
d. Meteorology
e. Oceanography
f. Combat strategies

Figure 4.1. Input categories.

Subsequently several of these input categories can be subdivided in input items.

	Input item	Input category
1.	Time	Time
2.	Geography	Place
3.	Slick properties	Chemical technology
4.	Oil properties	Chemical technology
5.	Sea water	Oceanography
6.	Weather	Meteorology
7.	Winds	Meteorology
8.	Waves	Oceanography
9.	Tides	Oceanography
10.	Currents	Oceanography
11.	Combat strategies	Combat strategies

Figure 4.2. Input items.

Each input item can in its turn consist of one or more input data.

The ever existing link between input data on the one hand and the physical processes (evaporation, dispersion, etcetera) on the other indicates that there are two possible approaches to the input problem:

(i) All available data are input. These data determine the physical processes that can be simulated by the model.

(ii) The required physical processes are selected. These selected physical processes determine the necessary input data.

The first approach (i) has been chosen on the basis of the following considerations:

- less insight in the physical processes is required from the operator
- a less complicated main simulation source program is required
- less economical on account of the fact that all collected data are not always necessary for the simulation
- no danger of non-available data to simulate a physical process

In figure 4.2. a list with eleven input items was given. Now a closer look at each item seperately is appropriate.

<u>1. Time:</u>

Input data: a. Date and time of commencement simulation
b. Date and time of termination simulation
c. Date of commencement interpolation*
d. Date of termination interpolation*
e. Output time-step in minutes

* See paragraph 4.3.

Comments : 1. The input of date and time is handled according to the Gregorian calendar. No conversion to an unit of time is necessary because the actual simulation has a time clock that corresponds to the Gregorian calendar (see paragraph 5.3.). The advantage of this method is that data files can always be directly adressed by the simulation without having to convert the data to a different time scale.

2. Time has to be specified up to one second. A more detailed specification is neither possible nor necessary.
3. The specification of an interpolation interval refers to the interpolation of time-dependent input data like wind-speed. See 7. Winds.
4. The output time-step determines the interval at which the simulation transfers results to data files.

<u>2. Geography:</u>

Input data:
- a. Name of map
- b. Minimum/maximum longitude/latitude of map
- c. Number of coastlines
- d. For each coastline: (i) Dike/Island/Mainland
  (ii) Latitude and longitude of the coastal points

Comments : 1. The objective of this input item is to make a visual display of the map on the screen. This item has not yet been incorporated in the simulation but most necessary preparations have been made. See appendix K.

2. The topography as used in MS4 (latitude/longitude) is tuned to maritime operators.
3. In the present-day simulation all input data like winds and waves are supposed to be valid for the whole area. However, these data are actually dependent of the position. More about this in paragraph 4.4..

3. Slick properties:

Input data: a. Longitude and latitude of the slick center
b. Spread acceleration, spread velocity and radius
c. Slick density, viscosity, surface tension, boiling point, vapor pressure and molecular weight
d. Several volumes

Comments : 1. Items a., b. and d. are always required
2. Item c. is only required when there is no question of an initial spill

4. Oil properties:

Input data: a. Oil density, viscosity and boiling point
b. Air-oil surface tension and molecular weight
c. Vapor pressure and the evaporable fraction
d. Residue density and emulsification constant

Comments : 1. All these data are required for an initial spill
2. It is possible to create a data base for regularly returning oil sorts

5. Sea water:

Input data: a. Sea water temperature, viscosity and density

Comment : 1. In case of future additions this can easily be taken care of by a simple change of form and source program. This applies for all items.

6. Weather:

Input data: Sunlight coefficient and air temperature

Comment : None

7. Winds:

Input data: a. Date and time
b. Direction
c. Backing/veering
d. Wind-speed

Comment : 1. Wind data can be given at any point of time.
2. MS4 uses a compass card with 360 degrees and direction is interpreted as the direction the wind blows from.

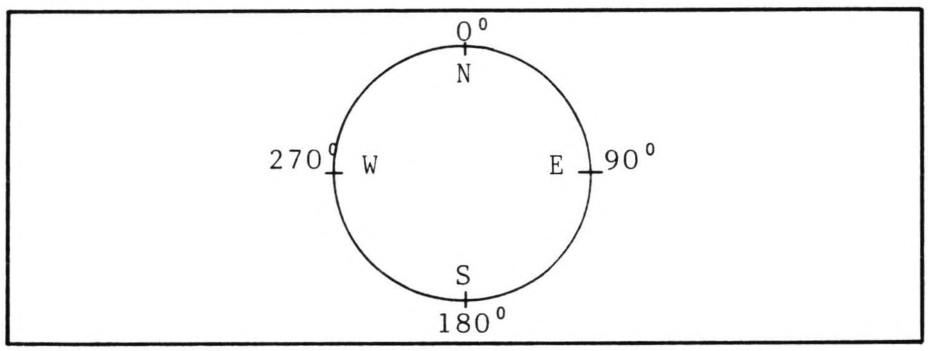


Figure 4.3. The compass card.

3. With the wind known at a limited points of time there is the necessity of interpolation (if necessary extrapolation) to calculate the wind characteristics at intermediate points of time. See paragraph 4.3.. Analogously the waves, tides and currents also need an interpolation routine.
4. The backing/veering item makes it possible to indicate in which direction the wind is changing.

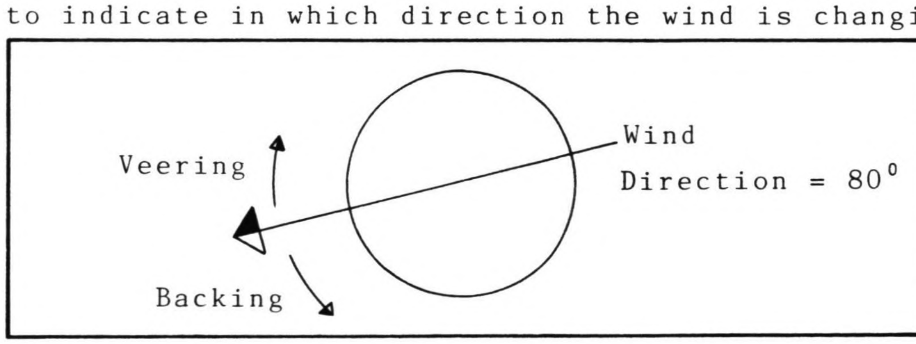


Figure 4.4. Backing and veering.

5. The wind-speed has to be input in meter/second. Although the method has not been used in MS4 it is possible by minor adjustments to the source program and extra forms to offer the operator various options for the input of wind data. E.g. input of wind-speed in m/s or in Beaufort. More about this possible MS4 extension in paragraph 4.2..
   Of course there are more input items where this possibility could be considered.

8. Waves:

Input data: a. Date and time
b. Wave height
c. Wave period
d. Direction

Comments : 1. Wave data can be given at any point of time.
2. MS4 uses a compass card with 360 degrees and the wave direction is illustrated by figure 4.5..

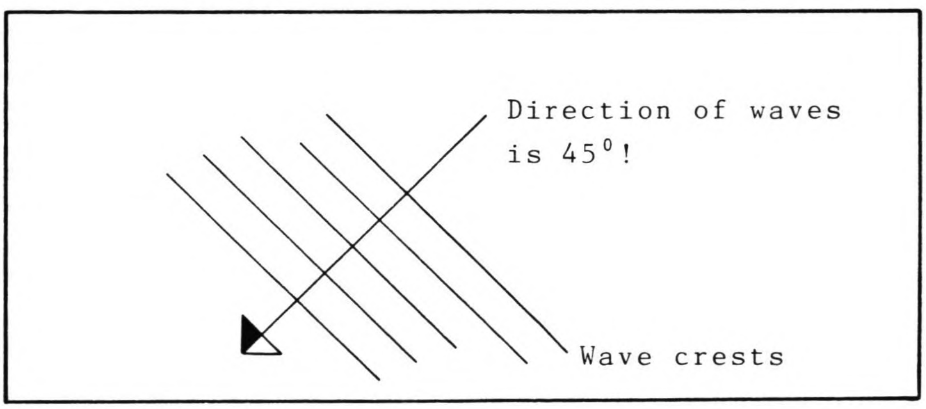


Figure 4.5. The wave direction.

9. Tides:

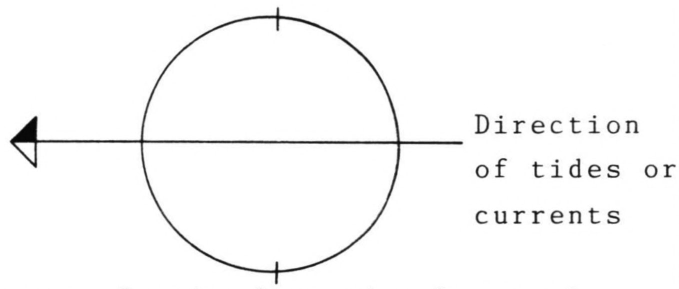
Input data: a. Date and time
b. Tidal speed
c. Direction

Comment : 1. Tidal data can be given at any point of time.
2. The direction is illustrated by figure 4.6..

10. Currents:

Input data: a. Date and time
b. Speed of current
c. Direction

Comments : 1. Current data can be given at any point of time.
2. The direction is illustrated by figure 4.6..



In this example the direction has to be entered as 90^0.

Figure 4.6. Direction tides/currents.

11. Combat strategies:

Input data: None! This item has been added in view of later addition of combat strategies.

4.2. Handling input data: FMS-11

The input of data has been handled by means of Digital's Forms Management System: FMS-11.
FMS-11 contains the tools for developing form applications.

Printed forms have been the most common tool for collecting and transmitting data in an orderly manner. FMS-11 software now brings the speed, convenience and accuracy of computerized processing to users who are used to printed forms.

Forms are designed by typing them directly onto the terminal screen. Neither layout charts nor a special forms design language are required.
FMS-11 associates constant data with the form, not with the application program, resulting in simplified application program maintenance and increased application program flexibility. Forms can later be modified without the need to recompile the application program. The form application programs have - like all other MS4 programs - been written in FORTRAN 77.

FMS-11 software has three main components for developing and executing form application programs:
- The Form Editor
- The Form Utility
- The Form driver

The Form Editor
The Form Editor has been used to design, modify and store form descriptions for video display. Short descriptions have been included in the form about individual fields and about each form as a whole.

The Form Utility

The Form Utility has been used to create and modify form libraries, to list the names of forms contained in each form library, and to produce object modules of form descriptions.

The Form Driver

The Form Driver is a set of subroutines that permits the application program to access forms that have been created with the Form Editor. Application programs access forms by issuing Form Driver calls that are imbedded in the task and are written in the source language of the task: FORTRAN 77. All Form Driver calls refer to specific forms and/or fields within forms using names that have been assigned during the form editing process. The Form Driver performs field and character validation for operator input based on the form definition. The Form Driver also responds to operator HELP requests by displaying appropriate help text associated with the form and field being processed.

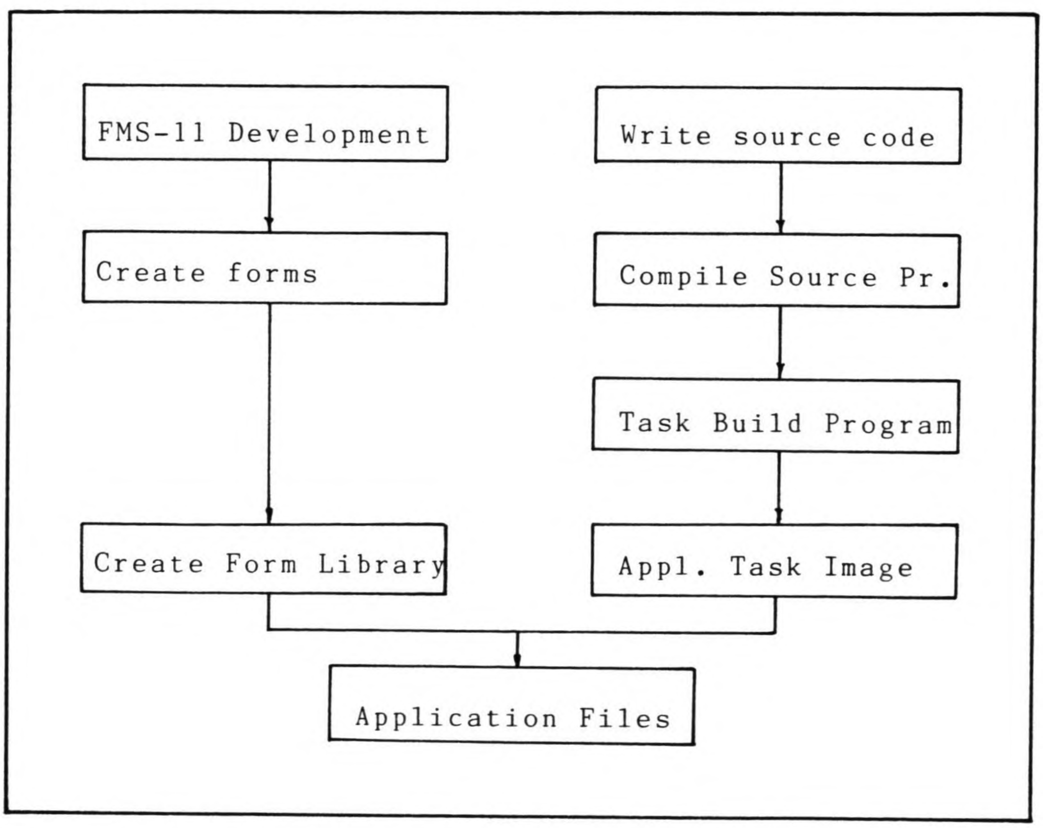


Figure 4.7. FMS-11 Development Cycle

The units of input data have been fixed in the forms. In case the available data have a different unit than required this can cause quite a problem.

Suppose the wind-speed has to be input in meter/second and all available data are in Beaufort. This would mean that all data must be converted to a different unit. This can turn out to be a rather time consuming business.

The design of forms with different units can effectuate that the operator can select an option indicating the units of the required input data. See figure 4.5..

```
Future MS4 possibility

Select one of the following options:
        →
        Input wind-speed in meter/second
        Input wind-speed in Beaufort

Make a selection and press the DO key:
```

Figure 4.8. The selection of wind input unit option.

Of course a minor addition to the source program is necessary. A processing module that converts the chosen unit to the default unit will suffice (in addition to the extra forms).

Although the method is not yet used in MS4 the programming structure has such been designed that the extension can be made easily.

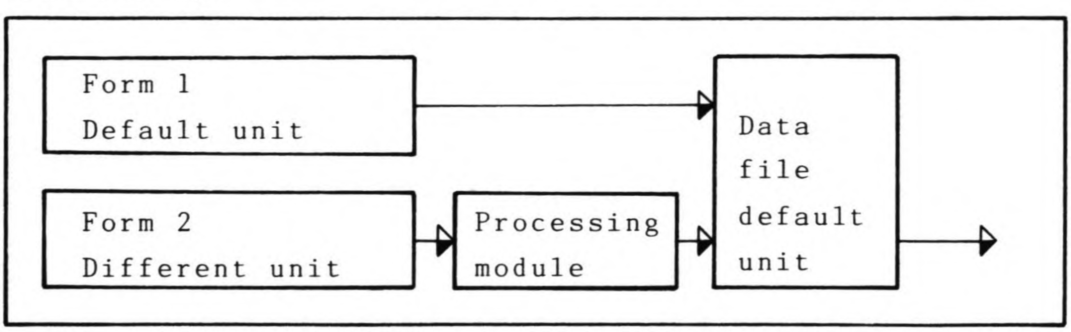


Figure 4.9. The use of various forms for the same input item.

A bit similar to the preceding method is the use of a data base. The reason to use a data base is to prevent the multiple input of the same data. Regularly used data are therefore stored in seperate files. This combination of data files is called a data base.

Now if the operator indicates that a certain file is needed then an additional processing module must make sure that the selected data file is transferred to the position where the simulation later expects its input data.

The method is clarified with an example.

Suppose the operator has to input the oil properties item. Then it would be very handy if a certain number of possibilities would be ready. E.g.:

1 Light Arabian Crude
2 Residual oil
3 Gasoline
4 Light Naphta
5 Other oil type

If one of the first four options is selected then the data file belonging to that specific option is transferred to the correct position for later use in the spill simulation.

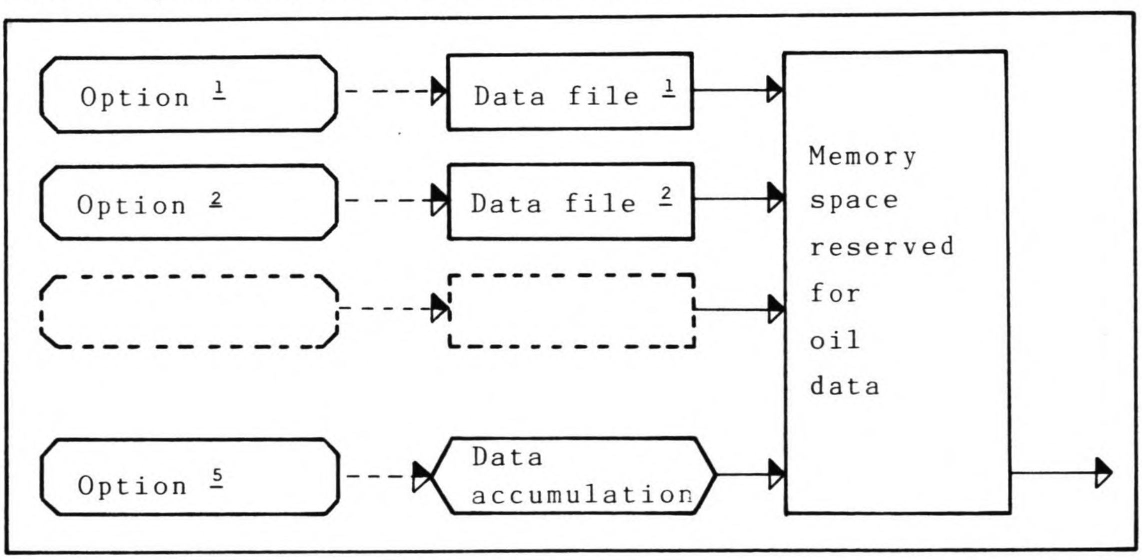


Figure 4.10. Data base operation.

4.3. The interpolation routine

There are four input items as function of the time:

- Winds
- Waves
- Tides
- Currents

These data are input at arbitrary points of time.
For the simulation data are also required for intermediate points of time. Therefore an interpolation routine is needed.

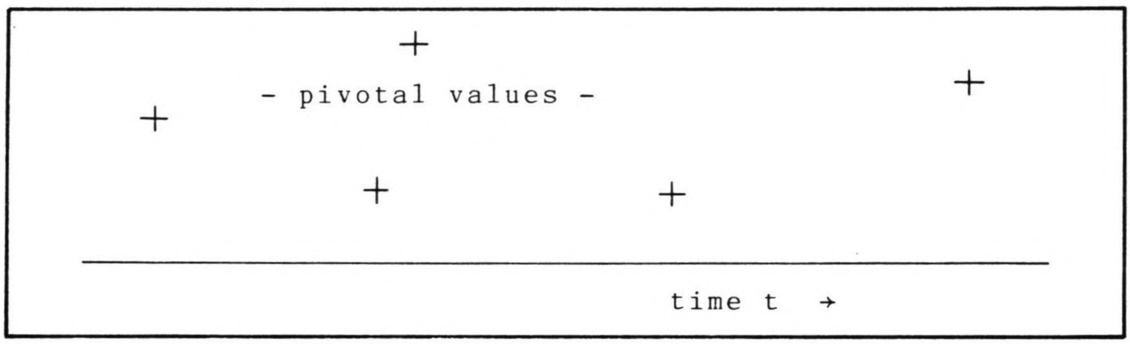


Figure 4.11. Data at various points of time.

SMOSS used a linear interpolation for the time-dependent data. This way the set of data can be schematized by a piecewise linear function; see figure 4.12..

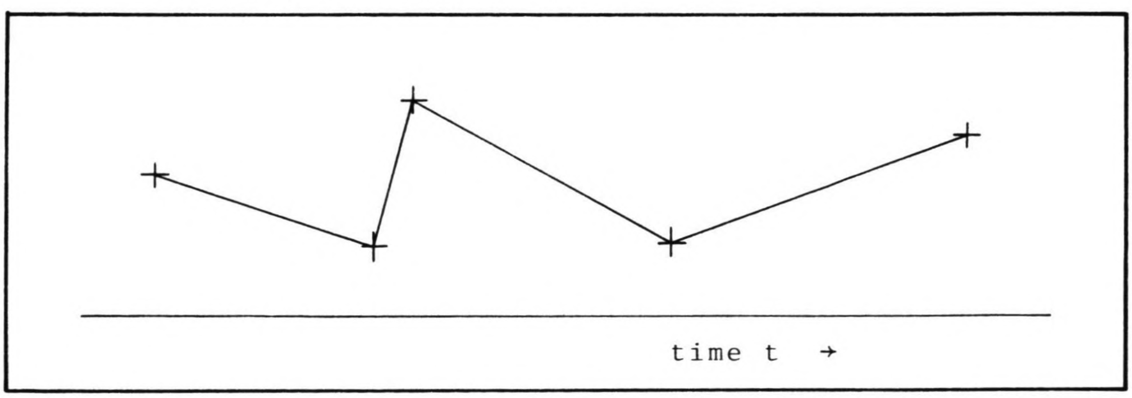


Figure 4.12. Linear interpolation SMOSS.

Now in SMOSS the interpolation took place during the actual simulation. There is no fault to find in this method but a primary objective of MS4 is the speeding up of run time.

Therefore the interpolation-module has been moved in front of the actual simulation instead of being a part of it. The advantages of this approach are:

- The actual simulation does not have to spend any computer run time on the interpolation.
- Assuming that the time-dependent data remain the same, the interpolation is only done once irrespective of the number of runs.

To prevent any interpolation during the actual simulation it is necessary to convert the set of data at arbitrary points into a tabulated list of data at regular time intervals. This way the time-dependent data can be input to the simulation as an array of values. See figure 4.13..

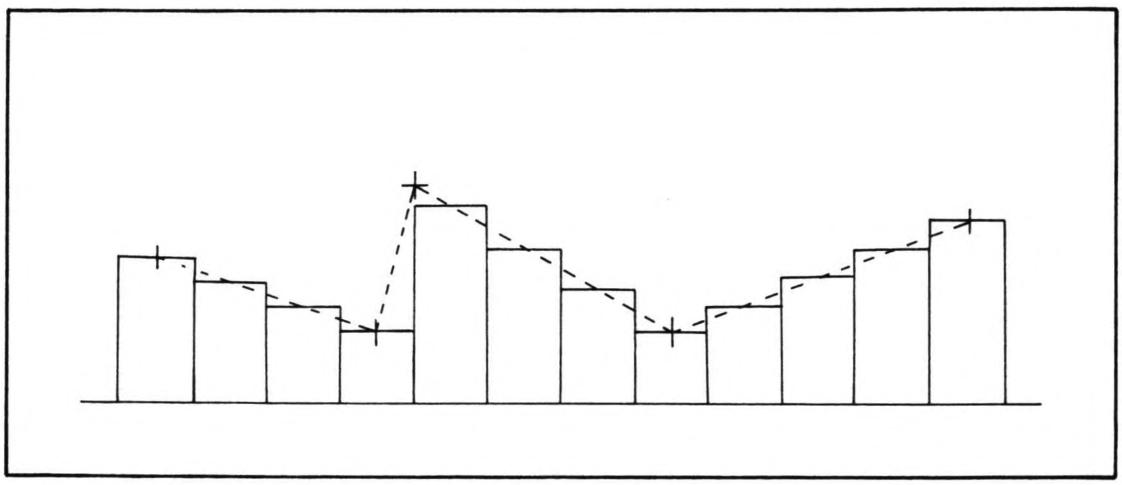


Figure 4.13. Values at a regular time interval.

It is clear that the time interval must be such that representative values of the data are given. A too large time interval would result in too large initial errors (see paragraph 7.4.). A too small time interval would lead to unnecessary computing of the interpolation-module. Based on the characteristics of the physical phenomena involved (winds, waves, tides and currents) and the time clock of the simulation (see figure 5.1.) an one hour time interval was chosen.

An example of such a tabulated list after interpolation of all data can be found in appendices I and J. These values are input as an array to the actual simulation.

To speed up computer run time the interpolation-module was uncoupled of the actual simulation. Futhermore values of data were then given at time intervals of one hour. To avoid becoming too inaccurate in the initial data, a different interpolation method has been used in MS4 (different from the SMOSS linear interpolation).

Objective is to approximate the data at arbitrary points by a smooth curve. This curve can in its turn provide the values at one hour intervals.

A widely used procedure for fitting a curve that yields a unique result depending only on the pivotal values is the method of least squares developed by Gauss.

For MS4 a least-squares curve-fitting program was developed that approximates n+1 pivotal values by a n^{th}-order polynomial equation. To set bounds to the computing the order of the polynomial equation is limited to 8.

The least-squares curve-fitting method results in the simultaneous solution of linear equations. For this problem the Gauss elimination method was selected. To improve the accuracy of this method patial pivoting was used: two rows of the matrix are interchanged so that the element with the largest absolute magnitude becomes the pivot element. Another reason why partial pivoting is useful is if a zero element appears on the major diagonal, then it will not be possible to divide the row by this pivot element.

In the data accumulation phase the operator has to specify the dates of commencement and termination of interpolation (see paragraph 4.1.). In the interpolation phase the curve-fitting is handled per day (the number of days follows from the dates of commencement and interpolation) based on the pivotal values of that specific day plus one pivotal value left and right of that specific day.
The reasons for this approach are:

- This approach is tuned to the time clock as used in the actual simulation; see paragraph 5.3..
- The total number of pivotal values can be enormous. An 8-th order polynomial equation curve -fitting would then be useless!

The method is illustrated in figure 4.14..

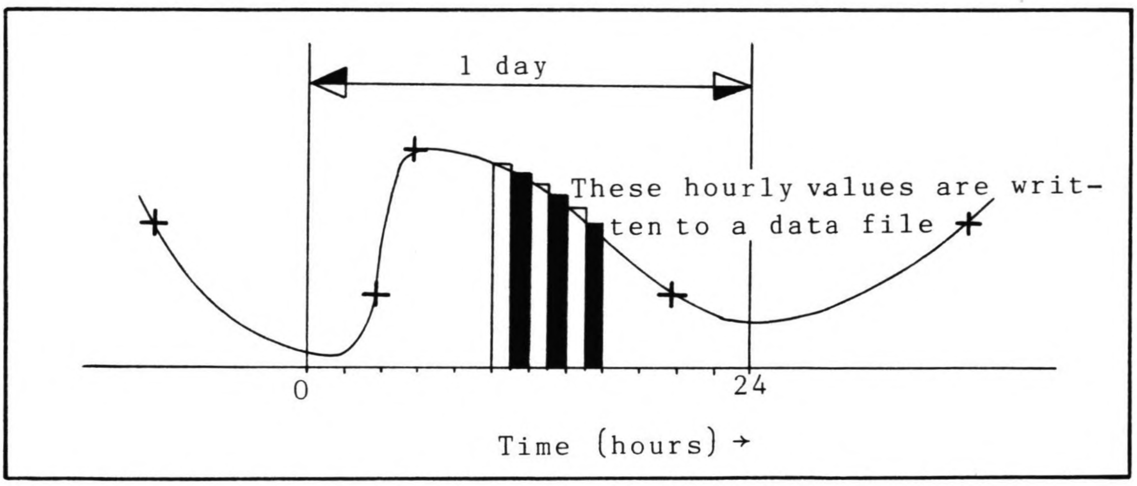


Figure 4.14. Curve fitting for one day.

The interpolation source programs for winds, waves, tides and currents can be found in appendix F.

4.4. Data dependent on the position in the horizontal plane.

In the present-day simulation model all time-dependent data - winds, waves, tides and currents - are assumed to be independent of the position in the horizontal plane. In other words all input and output data refer to one area.

However, it would be far more realistic to also allow input of these data, besides as function of the time, as function of the position in the horizontal plane.

The first requisite to such an approach is to input a geographical description of the whole area that can be of possible significance to the simulation run. See paragraphs 4.1. and 6.2.. Also see appendix K.

At this point there are two ways to incorporate the dependency of data on the position in the horizontal plane. These two ways are discussed in the following section.

(i) Divide the overall area into a number of sub-areas. The available data will be determining for this subdivision. These sub-areas can be specified by using the latitudes and longitudes as boundaries. Next number all sub-areas. See figure 4.15..

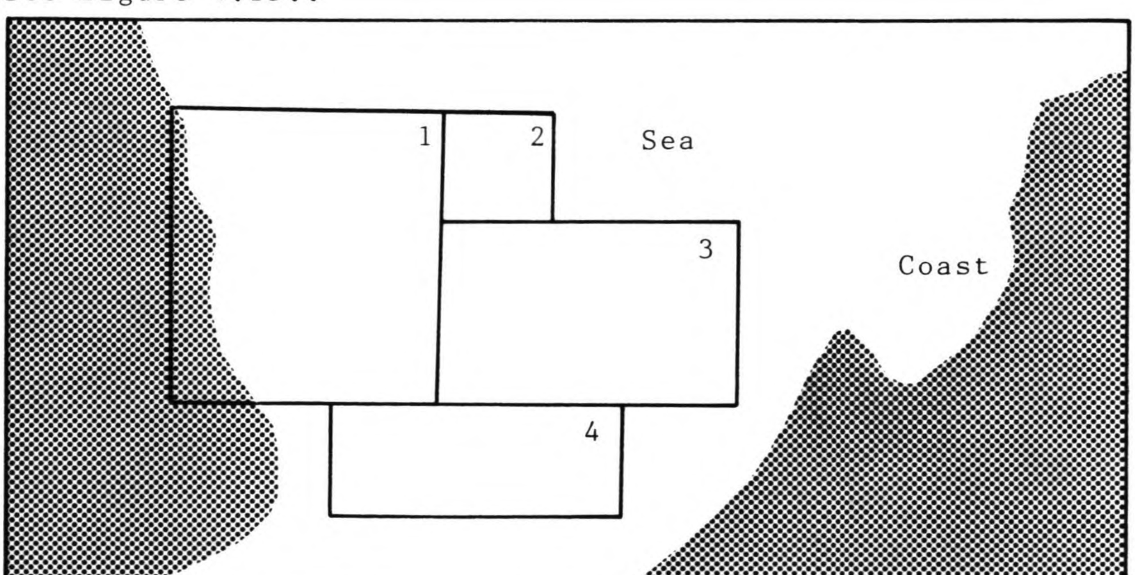


Figure 4.15. Subdivision of the area into sub-areas.

Now for each sub-area the input of data - as function of the time - can take place as usual. The processing of data into a tabulated list remains the same (see paragraph 4.3.). The number of tabulated lists equals the number of sub-areas.

During the simulation a check is made at each simulated hour: in which sub-area is the slick-center positioned? The data - necessary for the next hour of simulation - are then obtained from the tabulated list that goes with that specific sub-area.

(ii) Specify a point in the horizontal plane by entering its longitude and latitude. Time-dependent data can then be input for this specific point. The consequence of this approach is that - apart from the interpolation over the time - also an interpolation is required in the horizontal plane. Inverse proportionality of the distance from slick-center to point of input would then be an advisable method. However, a condition should be added to limit computing: limit the number of points that are to be interpolated in the horizontal plane. This can be achieved by taking only those points that lie within a certain distance from the slick-center. This is illustrated by figure 4.16..

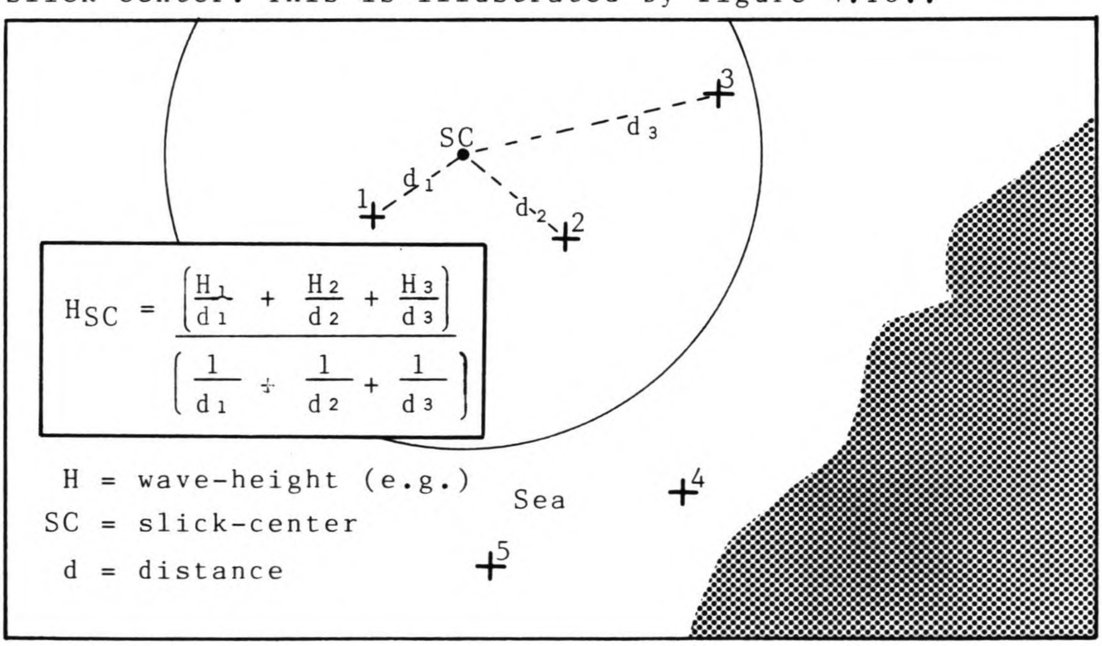


Figure 4.16. Interpolation in the horizontal plane.

Two methods that incorporate the dependency of data on the position in the horizontal plane have been discussed. Although both methods have their pros and cons, the first method (i) is recommended for future extension of the simulation. This choice is based on the following considerations:

- method (i) requires less computing than method (ii)
- the subdivision into sub-areas (i) instead of interpolating between points (ii) will usually give a better description of the real situation
- method (i) - in contrast with method (ii) - leads to discontinuities at the boundaries of the sub-area; however, these discontinuities are already introduced by tabulating values at one hour intervals

5. The actual simulation

5.1. Speeding up run time

A condition of outstanding importance is that the run time overall result should be within acceptable limits. Several items influence the run time of the simulation:

a. Hardware
b. Software

Hardware:

SMOSS used the IBM computer, a mainframe. Although dependent on simulation conditions, the run time can roughly be said to be 1 to 2 seconds for each hour of simulation.

In view of user friendliness the simulation was transferred to the PROFESSIONAL 350 (see appendix D), which is a micro-computer. This change of hardware alone results approximately in a speed reduction by a factor of 1000. This increases the run time for one hour of simulation to about 1500 seconds; a totally unacceptable computer run time.

This unacceptably long run time can only be decreased by making a considerable change in software.

Software:

SMOSS used the Continuous System Modeling Program (CSMP III) developed by the IBM company. CSMP turned out to be very rigid for handling the kind of problems encountered by SMOSS.

The rigid character of CSMP and the necessity of speeding up the run time has led to the choice of FORTRAN 77 as programming language. The reduction of run time, due to the change of programming language, is not exactly known.

To decrease the run time there are two alternatives left:
a. Simplify the theoretical description of the model
b. Use less time consuming subroutines

Theoretical description:

To simplify formulas or -more radically- to cut away parts of less importance to the simulation are very harsh methods. These methods will only be resorted to in case everything else fails to produce an acceptable run time.

Subroutines:

SMOSS uses two kinds of subroutines:
- AFGEN, an interpolation routine
- STIFF, an integration routine

The system subroutine AFGEN is a linear interpolation routine which hardly provides an opportunity to speed up run time.

However, the system subroutine STIFF, a sophisticated integration routine, is quite complex and by choosing a more simple routine it is possible to decrease run time significantly. This will probably cause less accurate results.

Considerations in selecting an integration method are summed up in the following paragraph.

5.2. Considerations in selecting the integration method

In general:

Selection of the best integration method for the simulation should not be made casually. The objective is to choose the routine that will provide the fastest execution while maintaining enough accuracy for the purposes of the simulation, and obtaining enough output data for easy interpretation of results. In general, as the complexity of the integration method increases, the computer time required for a single step also increases. On the other hand, the more complex methods are generally more stable, thereby permitting larger time steps. Thus, a number of tradeoffs between accuracy and run time are possible.

Basically there are two types of integration method:
a. The fixed-step integration method
b. The variable-step integration method

The fixed-step integration method:

This method uses a constant value for the integration step, as specified by the user, thus giving the user direct control of timing.

The variable-step integration method:

This method automatically varies the integration step as required to satisfy absolute and relative error criteria. For a given value of the integration step, speed of solution would be rated in the reverse order. Generally, however, the less sophisticated methods require smaller values of the integration step to achieve numerical stability.

Variable-step methods are sophisticated methods involving somewhat lengthy computation. They have the advantage of virtually ensuring a satisfactory solution, but at the cost of excessive running time if the error criteria have too stringently been set.

Selection:

Selection from among these several integration methods is simplified by first considering what output is required from the runs:
- Are both tabular and plotted output required?
- At what time interval is tabular output required?
- What should the overall duration of a simulation run be?

If the tabular output requires an interval very small compared to the duration of the simulation, it is possible that no significant advantage will be realized by selection of an integration method using variable-step-size adjustment. If the maximum step size is constrained by the relatively small output interval, use of the variable-step integration methods may increase solution time without significantly improving accuracy. In such a case, the less sophisticated integration methods should be considered.

5.3. The model time clock

Integration:

Several considerations in selecting the integration method have been mentioned. Apart from these considerations, attention should be paid to the handling of time in the model. After all, every integration in the model is done over an integration time step.

$$\text{Variable}\big|_{t+\Delta t} = \text{Variable}\big|_{t} + \int(\text{rate variable}).dt$$

So before selecting an integration method, a look at the time clock as used in the model is appropriate.

Time clock:

There are essentially two ways in handling the time:

1. Choose an initial point of time t_0 and express the time in an unit of time (seconds, minutes, hours) measured from this initial point of time t_0.

2. Tune the model to the most commonly used chronology, i.e. the Gregorian calendar.

ad 1. SMOSS used this method because it used the CSMP integration method STIFF, a variable-step method. In this way the run is started at 0 seconds and is terminated at - for instance - 36,000 seconds. The disadvantage of this method, being an advantage of the second method, will be discussed in the following section.

ad 2. In contrast to the first method which consists of one time loop (e.g. seconds) this method - as used by MS4 - consists of six loops. See figure 5.1..

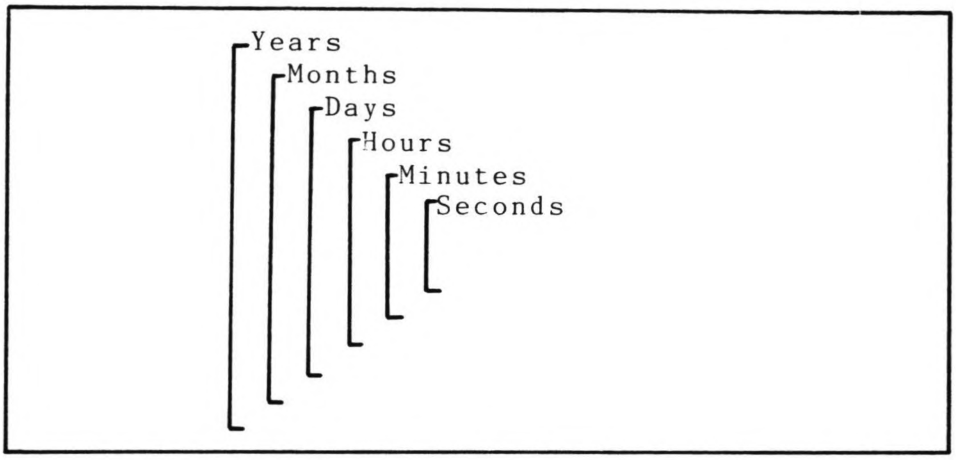


Figure 5.1. The MS4 time clock.

So the time scale of the simulated physical process determines in which loop the process will be placed. Slowly changing processes will be found in the more outer loops while the rapidly changing processes are placed in the inner loops.

Thus during simulation the values of the first three loops are as if it were a label indicating the date, and the values of the three inner loops are as if it were a label indicating the time.

The advantages of this method are:

- If all files containing time dependent data - like winds, waves, tides and currents - are labelled with date and time, the model can access these files directly without additional processing of the time scale.

- It is a clear, user friendly and unambiguous approach.

Of course the rate of change of physical processes can vary widely. Although this problem will be discussed more extensively in paragraph 5.4. it is sufficient to say that each process has two boundary limits dependent on its place in the loops.

5.4. Integration method and step size

As earlier mentioned, SMOSS used the highly sophisticated CSMP variable-step integration method STIFF. A routine tuned to processes with a highly fluctuating character.

In relation to the marine simulation the integration method STIFF - in spite of its high accuracy and excellent stability - has important disadvantages. These disadvantages are discussed in the following section.

Integration method:

STIFF is a highly sophisticated integration routine. It is a complex software package and not easily obtainable in the source language. This is required for application in MS4. Furthermore, computer run time is proportional to the complexity of the integration method. To avoid these disadvantages another - preferably simpler - method has to be selected, e.g. the trapezoidal rule.

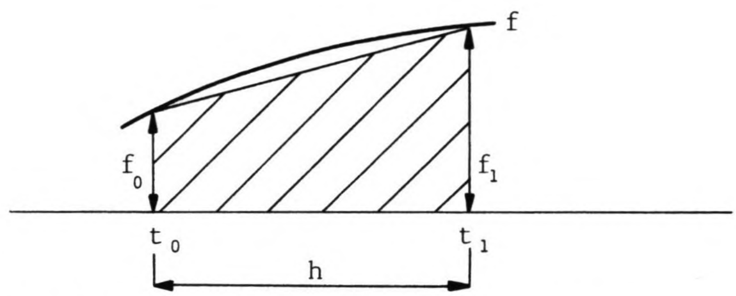
Integration step size:

SMOSS used a minimum integration step in the order of 3.6 E-4 seconds. In the MS4 view this step size is much too small and has been set at one second. This is the smallest step size possible within the model time clock (see paragraph 5.3.). The choice of an one second minimum step size has several consequences:

- improved computer run speed
- elimination of real numbers in the time clock resulting in a more economical use of memory space
- the loss of accuracy is estimated to be minimal relative to the model error. See paragraph 7.4. Accuracy.
- No physical process involved in this simulation requires a step size of less than one second

Integration step:

An important component of the simulation is the numerical integration of a system of coupled differential equations. Each numerical integration method has its own specific truncation error per step; see figure 5.2..



$$\int_{t_0}^{t_1} f(t)dt = \tfrac{1}{2}h(f_0 + f_1) + \text{truncation error}$$

$$\text{truncation error} = -\frac{1}{12}.h^3.f^{(2)}(\xi);\ \xi \in (t_0;t_1)$$

Figure 5.2. Example truncation error: the trapezoidal rule.

The general expression for the truncation error per step is:

$c.h^n.f^{(p)}(\xi)$ with c = constant (real number)
h = step size
n = power of h (positive integer number)
f = integral curve
p = order derivative of f
ξ = point in integration interval

To guarantee a certain accuracy, an error criterium can be set which the truncation error per step may not exceed. In SMOSS all differential equations are integrated with the same integration method and step size. Therefor the set error criterium will be reached first by the most rapidly fluctuating process. However, all other processes are forced to integrate with the same step size irrespective of their own truncation error per step. This is a very uneconomical approach indeed!

The ideal solution to this problem would be the complete uncoupling of all differential equations and thus making all processes completely independent. However this is not possible in its totality but partially it can and has been done. The simulation can be divided in three blocks.

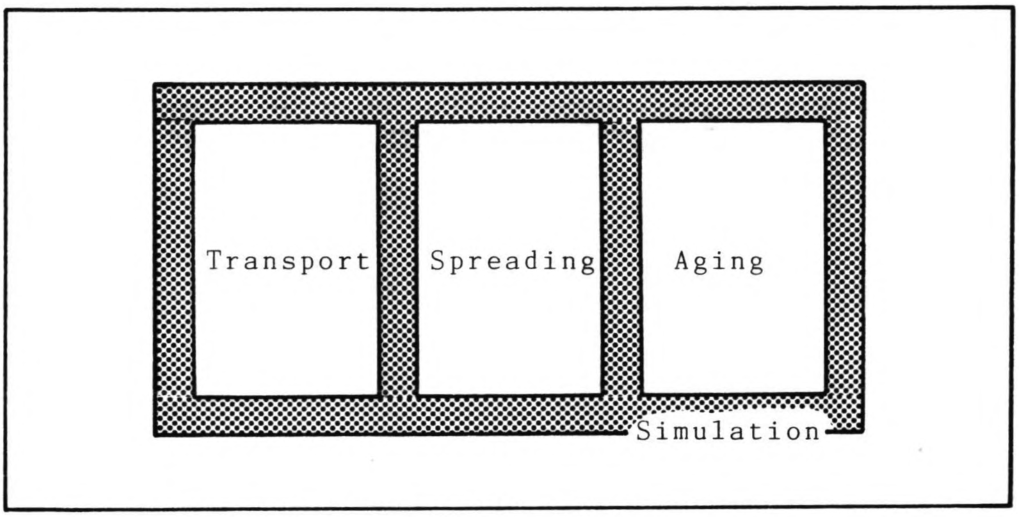


Figure 5.3. Subdivision simulation processes.

The result of this uncoupling into three seperate modules is that each method can choose its own integration method and step size. Although the problem of the integration step has remained the same, it only forces one module to integrate with the chosen step size. The two remaining modules are safeguarded against an unnecessarily small integration step size.

5.5. Transport, spreading and aging

The simulation consists of three physical processes:

- Transport
- Spreading
- Aging

Transport:

The transport process is completely independent of the spreading and aging processes. The transport process calculates the movement of the slick center - the slick is assumed to be circular - in the horizontal plane.

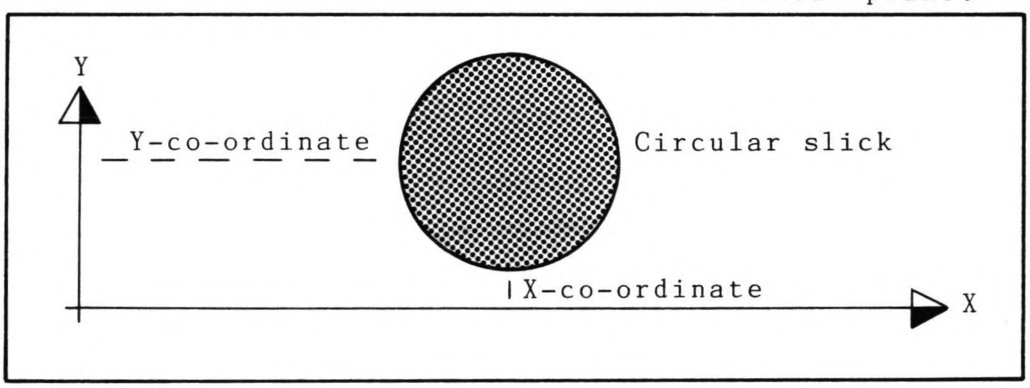


Figure 5.4. The circular slick in the horizontal plane.

The slick transport is based on wave, wind, tidal and current data. These data have been tabulated for each hour of the day (the why and wherefore can be read in paragraph 4.3.). These values have been summed and have been witten to the slick velocity data file VELOX.DAT.

The slick velocity, as tabulated in VELOX.DAT, has to be integrated to calculate the slick transport.

The problem of numerical integration is the numerical evaluation of a definite integral

$$J = \int_a^b f(t)dt$$

where a and b are given and f(t) is a function given by a table of values.

These values are tabulated with one hour intervals.

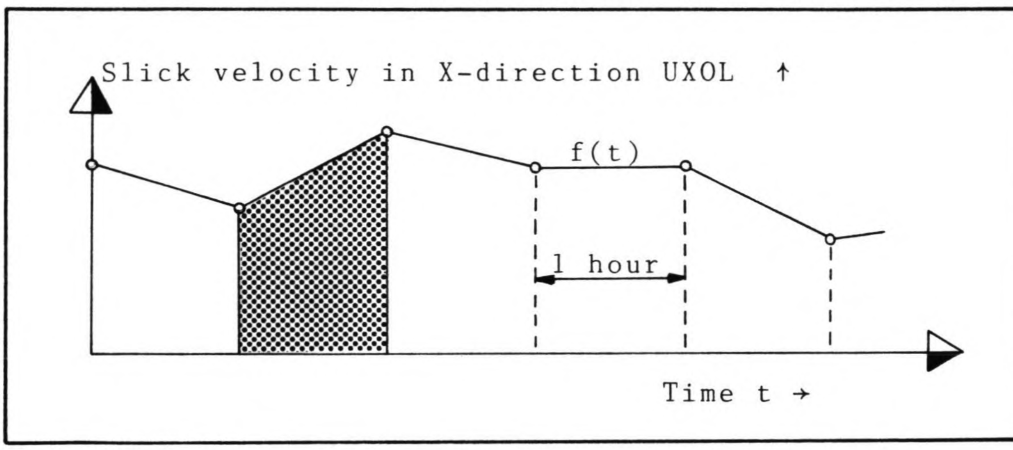


Figure 5.5. The trapezoidal rule.

As already discussed in paragraph 4.3., the function f(t) is approximated by a piecewise linear function. This approach of the interpolation effectuates that the trapezoidal rule will suffice as numerical integration method.

$$X\Big|_{t+\Delta t} = X\Big|_t + \frac{UXOL\big|_t + UXOL\big|_{t+\Delta t}}{2} \cdot 3600; \quad \Delta t = 3600 \text{ seconds}$$

X = X-co-ordinate slick

UXOL = slick velocity in X-direction

Although integration over an one hour period is possible, it can be recommended - at the cost of computer run time - to take a smaller integration step in case the user requires output at a smaller output interval than one hour (e.g. see appendix I). However, this dces not influence the method of integration, only the step size.

Spreading/aging:

Although spreading and aging are entirely different physical processes, with regard to the selection of the integration method they can be discussed together.

Irrespective of the integration method selected for spreading and aging, the integration method must correspond to the the simulation time clock as described in paragraph 5.3. The model time clock.

Because of this time clock there are two condions:

- The spreading and aging process should be placed somewhere in the six time loops. For both processes the most inner loop has been selected. It is expected that this time loop will correspond the best to the time scale of these processes. By selecting the most inner loop the lower limit is set at one second and the upper limit at sixty seconds, i.e. one minute.

- Because the upper limit is set at sixty seconds it should be taken into account that the sum of integration steps must always end at sixty seconds.

As has been mentioned in paragraph 5.2. Considerations in selecting the integration method, there are two types of integration methods: fixed-step and variable-step integration methods.

For the MS4 purpose a hybrid integration method has been selected. It could also be called a regulated variable-step integration method. Based on an error criterium the chosen method determines the step size, but possible step sizes have been prescribed.

Spreading:

Radial acceleration = ACC
Radial velocity = UR
Slick radius = R

Two variables of the spreading process require integration:

$$UR\Big|_{t_1} = UR\Big|_{t_0} + \int_{t_0}^{t_1} ACC\Big|_{t_{0/1}} .dt$$

$$R\Big|_{t_1} = R\Big|_{t_0} + \int_{t_0}^{t_1} UR\Big|_{t_{0/1}} .dt$$

Method of determination step-size:

The integration step-size will be determined by the value of the second derivative of the radial velocity: UR".
If the value of UR" should at a point of time exceed a set value the step-size is decreased (if possible, the minimum step size is one second). The reasoning is also valid the other way round. Should UR" at a point of time be less than the set value, the step is increased (if possible, the maximum step-size is sixty seconds).

Point of time determination:

With the exception of the first minute of the simulation the step-size is always determined at the beginning of each minute: Δt_{det} = 60 (seconds). So the ratio of process computation to step-size determination computation can vary, dependent on the step-size $\Delta t_{sp/ag}$, from 1 : 1 to 60 : 1.

Starting procedure:

The first minute of the simulated time period always uses a step size of one second. The possible step-sizes being 1, 5, 20 and 60 seconds the step-size determination has been designed as such that the step-size can never jump from one second to twenty or sixty seconds. Always the adjacent step-size.

What is possible is that the simulation commences with a step-size Δt_{sp} of one second, in the second minute a Δt_{sp} of five seconds, the third minute a Δt_{sp} of twenty seconds and in the fourth minute a sixty seconds step size.

So long as the error criterium is not exceeded, the simulation can proceed with a step-size Δt_{sp} of sixty seconds.

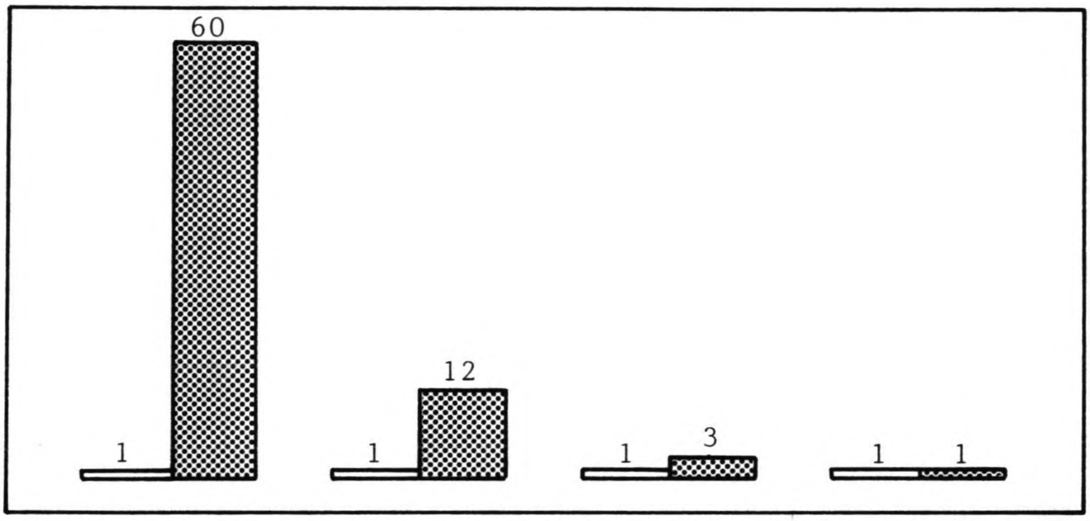


Figure 5.6. Ratio $1/\Delta t_{det}$: $1/\Delta t_{sp}$ for Δt_{sp} = 1, 5, 20 and resp. 60 s.

Aging:

The method of reasoning for the aging process is rather the same as for the spreading process with this exception that there are more variables that require integration and that the step-size Δt_{ag} of the integration is determined by the second derivative of the volume oil at the surface VOILSF, being the most representative variable for the aging process.

It is problably worthwhile to examine if the step-size determination can be eliminated. The alternative would be to start the simulation in the first simulated minute with a step-size of one second, the second minute a 5 seconds step, the third minute a 20 seconds step and for the rest of the simulation a 60 seconds step-size. The decisive answer to this suggestion will be gotten by accuracy conditions; see paragraphs 7.4 and 7.5..

The selection of an integration method is now rather limited by all these conditions. After careful consideration the trapezoidal rule has been selected. Some of these considerations are:

- it is a simple method which is reflected in the required computer run time
- no software package is required
- unity; transport also used the trapezoidal rule
- no starting method is required

Summary:

Integration method : The trapezoidal rule

Truncation error per step : $-\frac{1}{12}.h^3.f^{(2)}(\xi)$

Fixed integration step sizes of spreading and aging process: $\Delta t_{sp/ag}$ = 1, 5, 20 or 60 (sec)

Step size determination : The most logical course of action is to couple the step size determination to the truncation error. The truncation error is a function of the second derivative of f, so what would be better than to base the error criterium on the value of the second derivative of f.

6. Output

6.1. Handling output data

MS4 data can be grouped in four different categories:

a. Initial data : At the begin of the simulation all initial data are accumulated and stored in seperate data files. These data relate to the time period of the simulation and interpolation, the geography, slick properties, oil properties, sea, weather, etcetera.

b. Processed data: These are the interpolated wind, wave, tidal and current data along with the resulting slick velocity data.

c. Resulting data: These data result directly from the simulation. There are three categories:
- Transport data
- Spreading data
- Aging data; see figure 6.2.

d. Internal data : These data have fixed values within the simulation. The operator can only change these data by changing the source program.

A few remarks about the MS4 data:

- At the data accumulation phase the user can specify the simulation time step for writing data to all files. This can be any real number between one minute and sixty minutes. This timestep (default five minutes) can best be tuned to the period of simulation in order to avoid huge data files.

- All data files are located on the Winchester disk. In view of computer run time this is rather a must because read/write operations to a floppy disk are relatively slow.

- Each data file has a name representative for its contents, e.g. SPREAD.DAT stands for the spreading data file. All data files have as filetype .DAT.

- Each data file is headed by the full name of the file as well as date and time of storage. This enables the operator to distinguish different runs. The distinguishing of different data files is also made easier by the version number. The latest file has the highest version number.

- In order to conserve free storage capacity, the operator should now and again purge all old data files.

- No file protection is used. So the user should be attentive for his own sake when purging or deleting data files.

- During the simulation data sets are stored using six communication channels. Every channel requires quite a lot of memory but the alternative is to use one channel resulting in one file with all data. This would not improve the clarity of arrangement.

- All data that are output as function of the time are stored together with the time and date belonging to that data.

- Data files are never overwritten.

Devices that output data are:
- A printer
- A visual display unit

The present-day simulation model only uses the printer to output data. This is namely the only way to get a hard copy of the files.

MS4 can be schematized in three main modules:
- Data accumulation
- Data processing
- Simulation

In principle the data resulting of these modules can only be output after complete execution. See figure 6.1..

The lay-out of each file is such that a file print will perform the output.

Based on the fact that results can only be viewed after complete execution of the simulation, there can not be any assesment or intervention during simulation. However an "simulation hour" takes roughly four seconds so this can not be too big a problem.

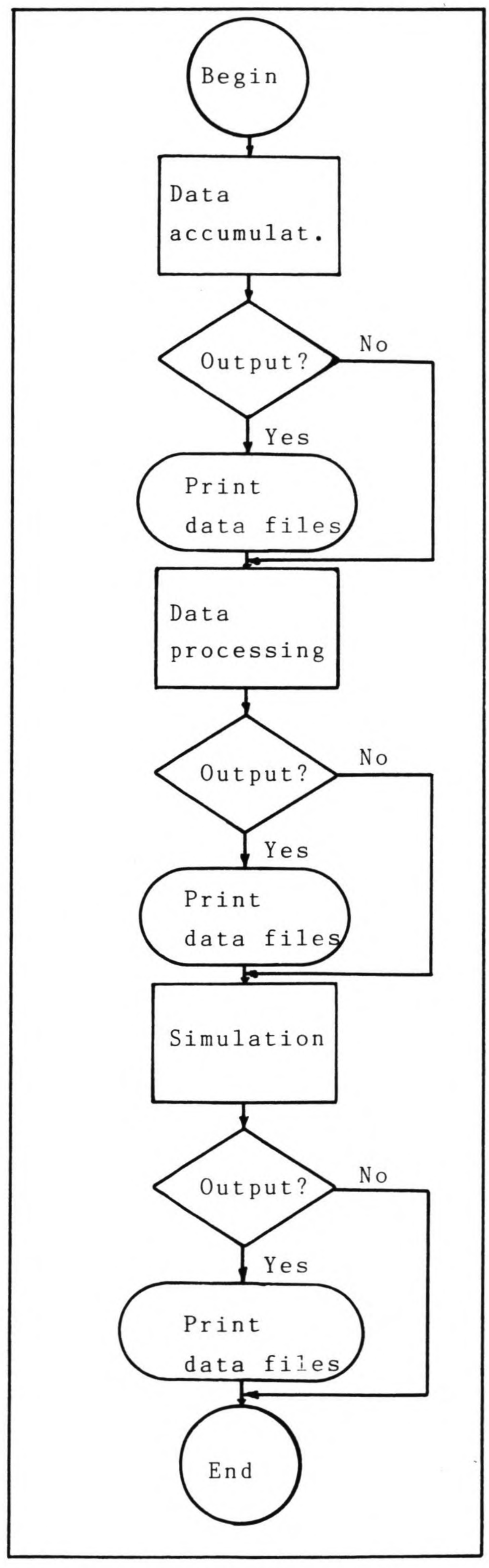


Figure 6.1. Flowchart MS4

Aging data: Part 3

Time: 15:03:07 Date: 03-DEC-84

Date {year-mt-dy}	Time {hr:mn:sc}	Vevapo {m^3}	Vseair {m^3}	Vdispe {m^3}	Vdisso {m^3}
1984- 9-19	0: 0:00	0.0	0.000	0.0	0.0000
1984- 9-19	0: 5:00	0.5	0.126	37.8	0.0036
1984- 9-19	0:10:00	3.1	0.251	75.4	0.0072
1984- 9-19	0:15:00	9.8	0.377	113.0	0.0108
1984- 9-19	0:20:00	19.7	0.502	150.5	0.0143
1984- 9-19	0:25:00	31.2	0.626	187.9	0.0179
1984- 9-19	0:30:00	43.6	0.751	284.7	0.0214
1984- 9-19	0:35:00	56.6	0.874	469.9	0.0250
1984- 9-19	0:40:00	70.0	0.997	506.7	0.0285
1984- 9-19	0:45:00	83.6	1.119	543.5	0.0320
1984- 9-19	0:50:00	97.5	1.241	580.1	0.0355
1984- 9-19	0:55:00	111.5	1.363	616.6	0.0389
1984- 9-19	1: 0:00	125.8	1.484	653.0	0.0424
1984- 9-19	1: 5:00	140.1	1.601	688.0	0.0459
1984- 9-19	1:10:00	154.6	1.717	722.9	0.0493
1984- 9-19	1:15:00	169.2	1.833	757.7	0.0528
1984- 9-19	1:20:00	184.0	1.949	792.5	0.0562
1984- 9-19	1:25:00	198.8	2.065	827.1	0.0596
1984- 9-19	1:30:00	213.8	2.180	861.6	0.0630
1984- 9-19	1:35:00	228.8	2.295	896.1	0.0664
1984- 9-19	1:40:00	244.0	2.409	930.4	0.0698

Figure 6.2. An example of a data file print.

6.2. Future output presentation

Although MS4 only uses the printer to present output, the use of visual display of output has been anticipated.

- In the data accumulation phase a program has been added to make the plotting of a map on the screen possible. The FORTRAN program that does the plotting of the map on the screen is not yet available. However a BASIC program is available; see appendix K. A hard copy is easily made; the operator only has to press the SCREEN PRINT key. In this manner the operator can follow the fate of the spill on a screen image of the map.

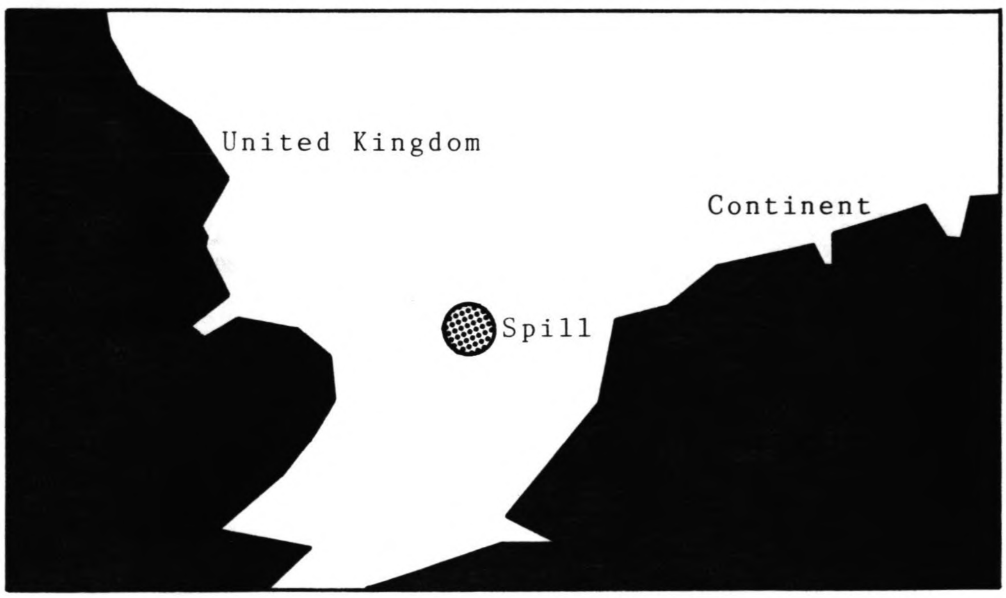


Figure 6.3. Future presentation fate oil spill.

- All data resulting from processing and simulation are written to files linked with date and time. A future program will allow the user to specify the variable and the time interval producing a graph on the screen.

7. Program testing

7.1. Introduction

To check if all processing is done correctly, the program is run with test data for which the results are known.

Due to the lack of complete SMOSS runs and the amount of work involved, the program testing has been limited to two runs. The first run simulates a spill over a period of one hour and fourty minutes, the second run over a period of four days and four hours. Although the certainty of correct processing increases with an increasing number of test runs, these two runs nevertheless give a fairly good indication of the correct processing of MS4.
A point in favour for these two runs is their different time scale.

Accuracy and efficiency have been chosen as main elements of a comparison between SMOSS on the one hand and MS4 on the other.

The MS4 test run results, which all are variables as function of the time, have been plotted in graphs together with SMOSS results thus giving an excellent view of the accuracy achieved.
A list of all plotted variables - definition, notation and unit - is given in figure 7.1. on the following page.

Definition	Notation	Unit
X coordinate slick	X	meter
Y coordinate slick	Y	meter
Radial acceleration	ACC	meter/second^2
Radial spread velocity	UR	meter/second
Radius	R	meter
Height layer	HL	meter
Surface slick	SURF	kilometer^2
Oil density	RHOO	kg/m^3
Oil viscosity	NUO	meter^2/second
Boiling point	BP	^{0}Celsius
Net surface tension	SIGMA	N
Vapor pressure x molecular weight	PM	
Total slick volume	VTOTAL	meter^3
Water volume in slick	VWATER	meter^3
Volume oil at the surface	VOILSF	meter^3
Volume oil accent	VACCNT	meter^3
Emulsified oil volume	VOILEM	meter^3
Evaporated oil volume	VEVAP	meter^3
Volume oil sea-air exchanged	VSEAA	meter^3
Dispersed oil volume	VDSP	meter^3
Dissolved oil volume	VDSS	meter^3

Figure 7.1. List of all plotted variables.

7.2. First test run: 100 minutes

With regard to SMOSS results reference is made to:

- SMOSS
  A Simulation Model for Oil Slicks at Sea
  Appendix E
  Paragraph E.3. 6000 seconds run

With regard to MS4 results reference is made to:

- Appendix I (seperate volume)

Two curves are plotted in each graph.
The SMOSS curve: - - - - - - - - - - - - - - - -
The MS4 curve : ————————————

The vertical scale of the following graphs are mostly linear. In some cases however a logarithmic scale was used.

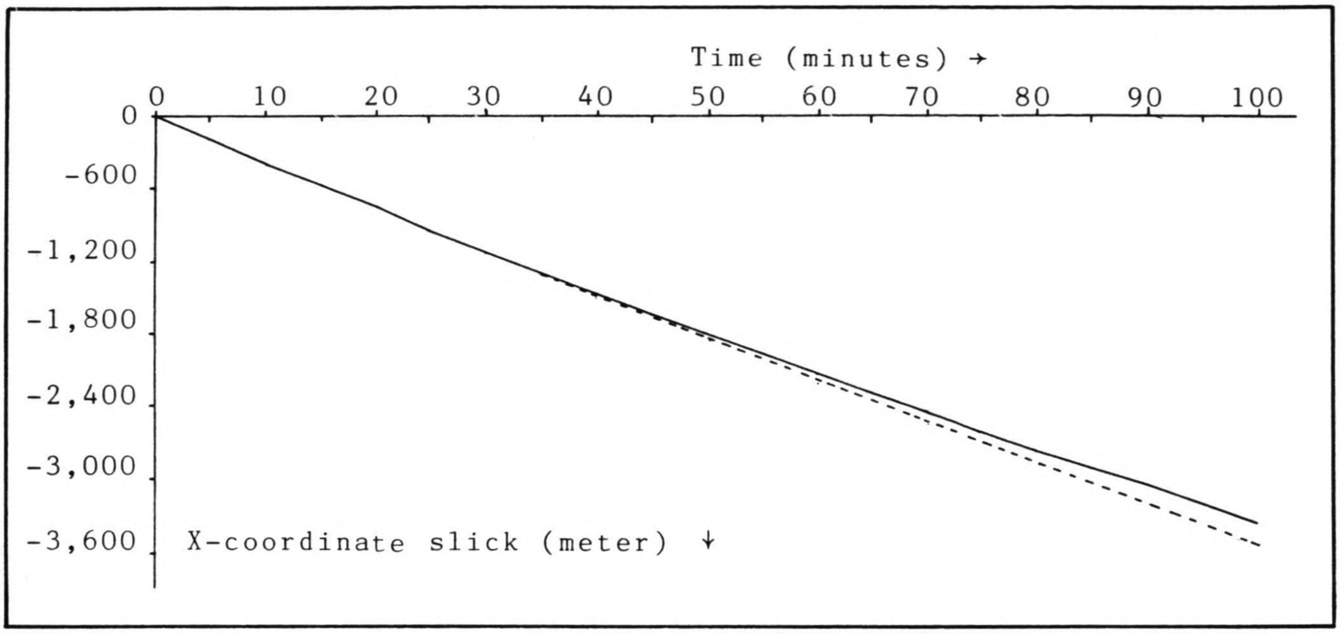


Figure 7.2. The SMOSS and MS4 curve of X.

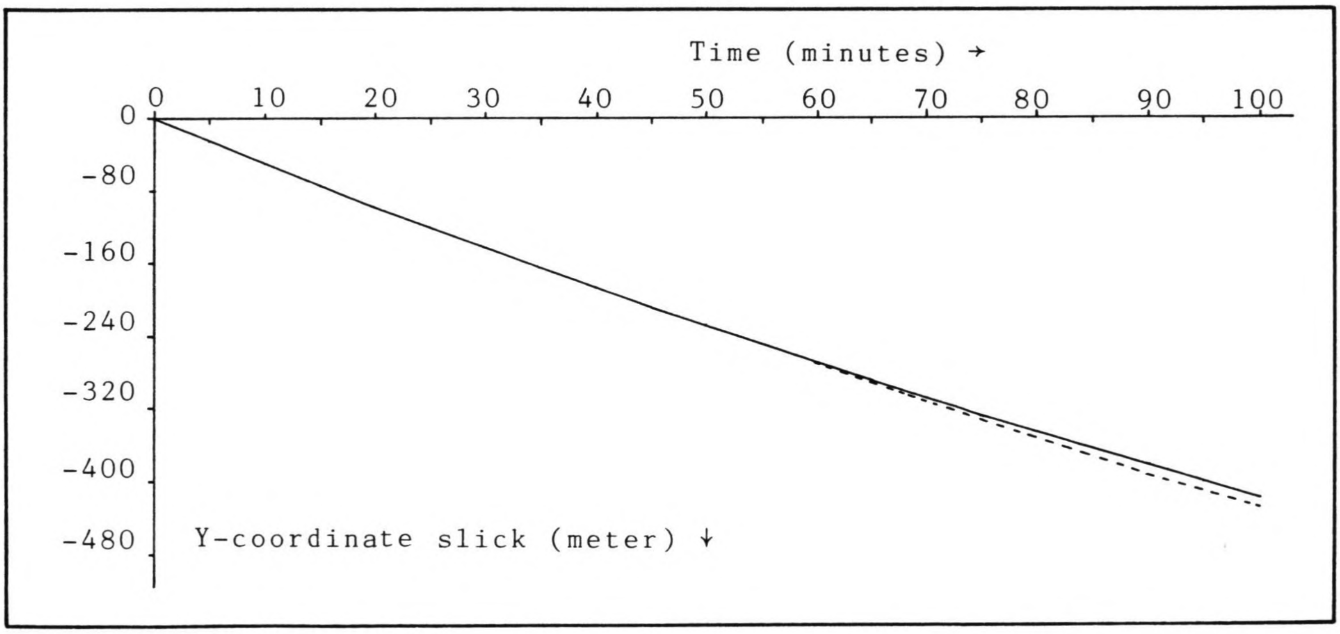


Figure 7.3. The SMOSS and MS4 curve of Y.

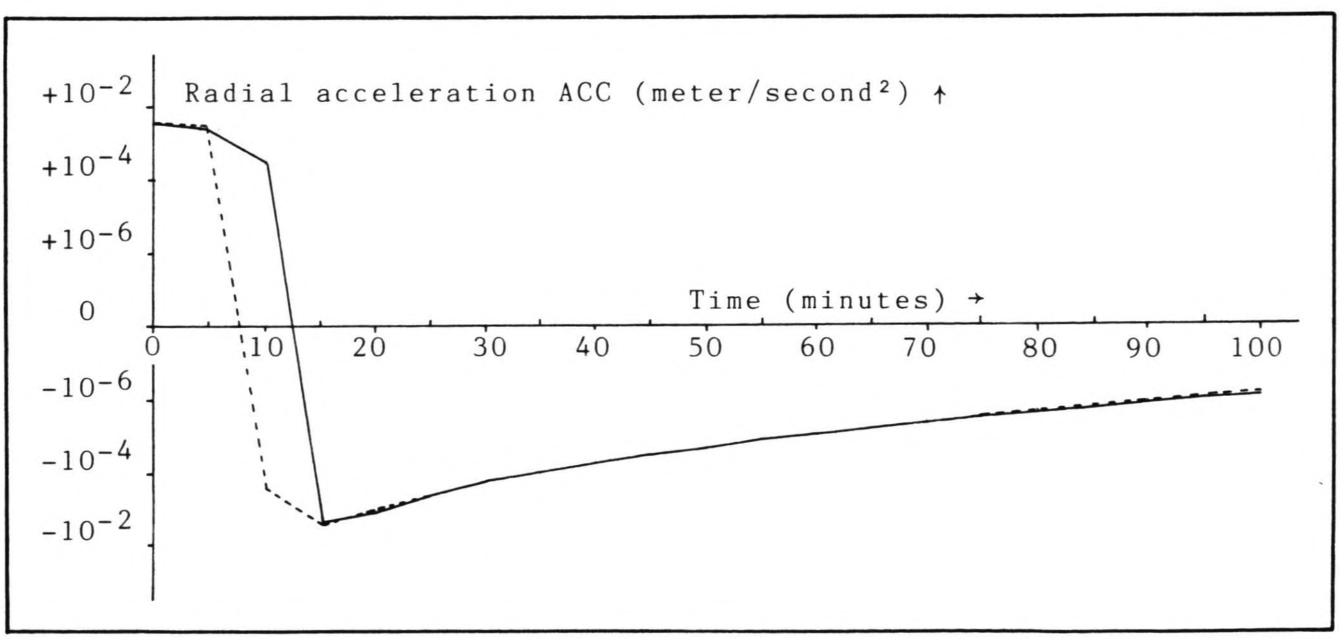


Figure 7.4. The SMOSS and MS4 curve of ACC.

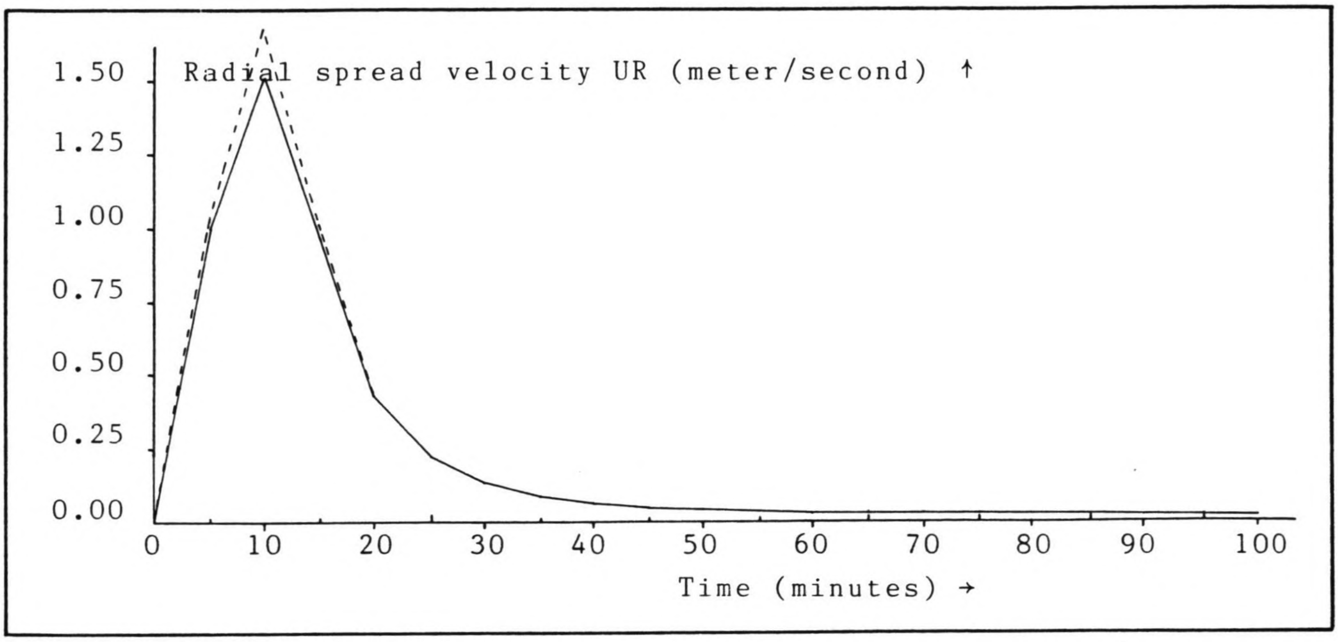


Figure 7.5. The SMOSS and MS4 curve of UR.

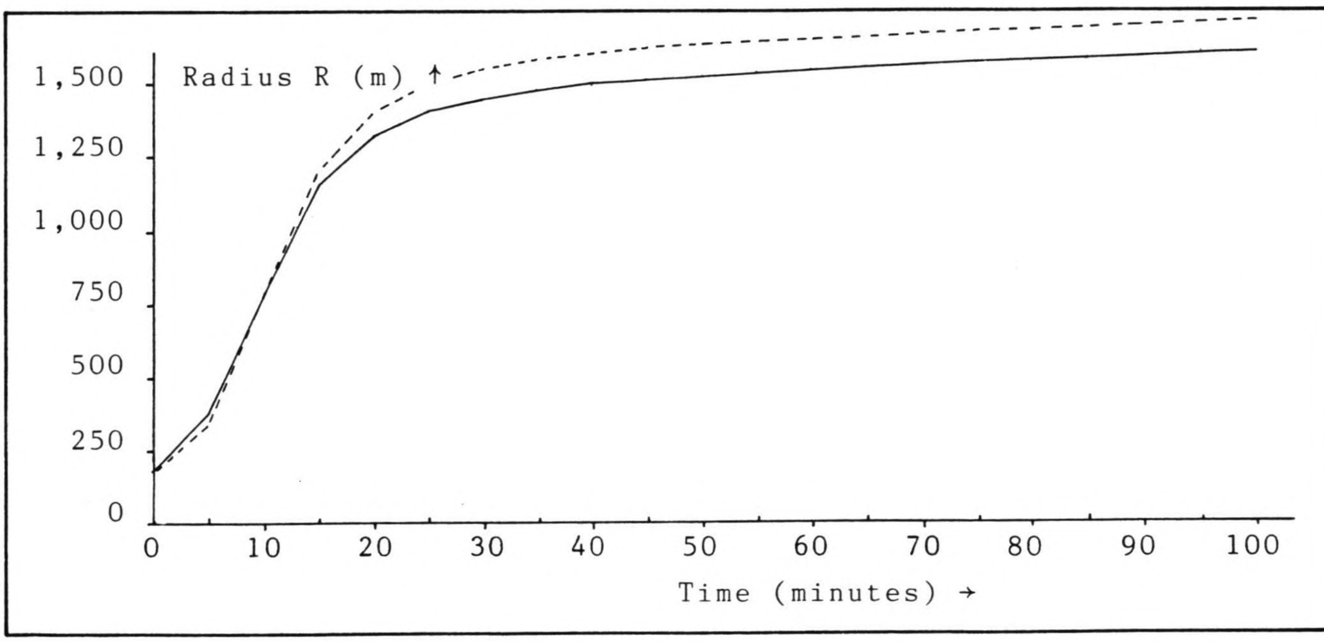


Figure 7.6. The SMOSS and MS4 curve of R.

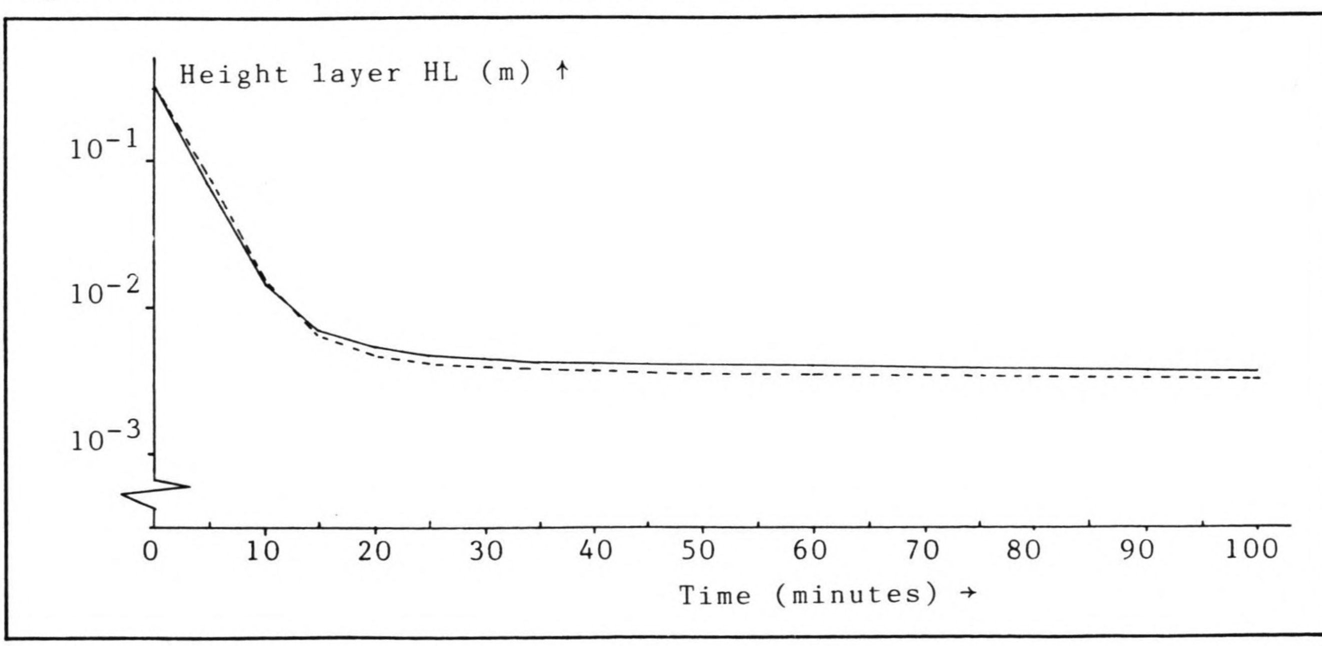


Figure 7.7. The SMOSS and MS4 curve of HL.

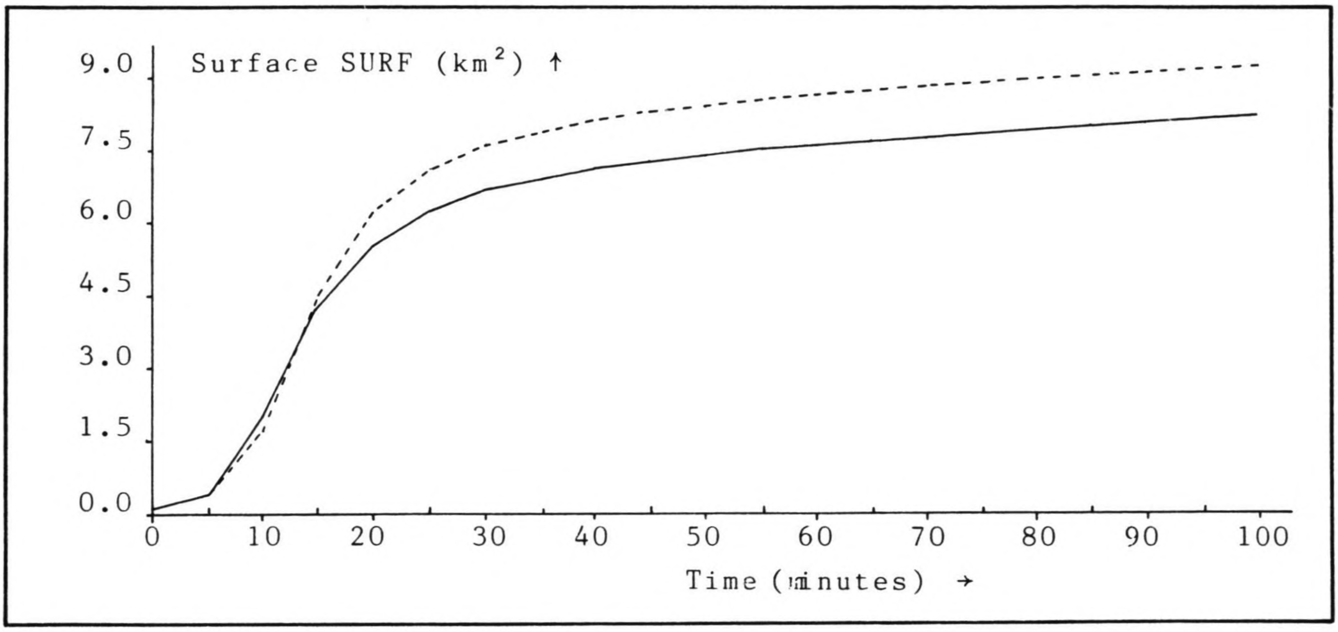


Figure 7.8. The SMOSS and MS4 curve of SURF.

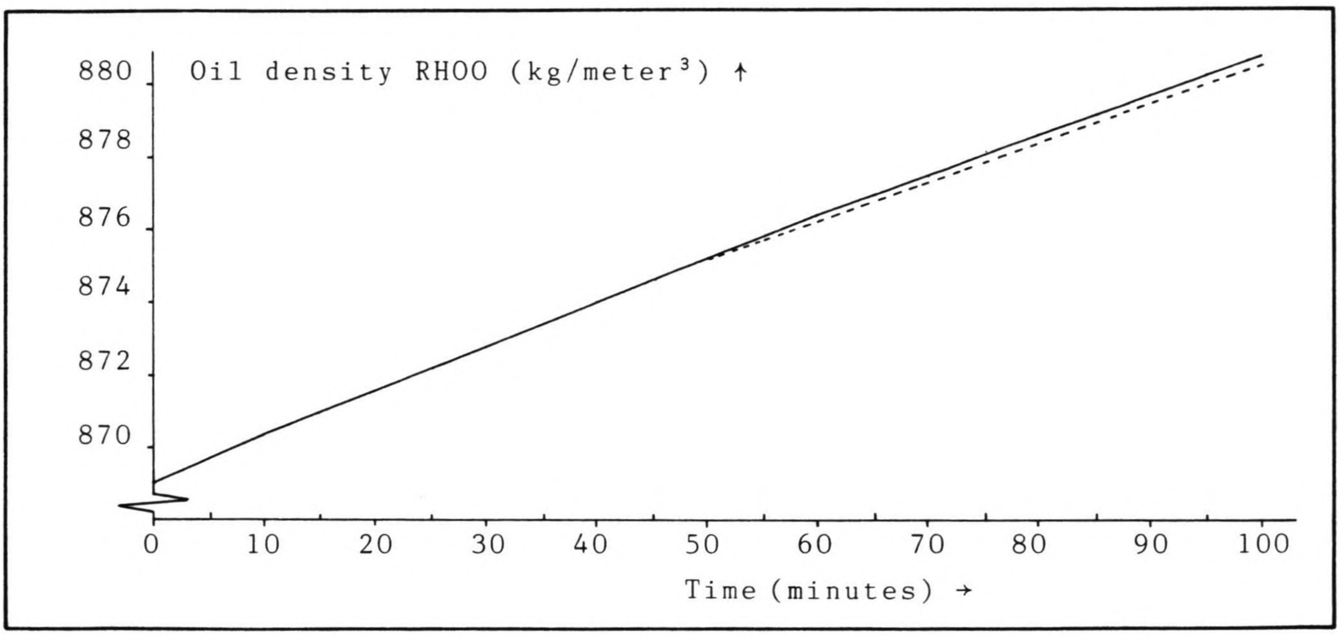


Figure 7.9. The SMOSS and MS4 curve of RHOO.

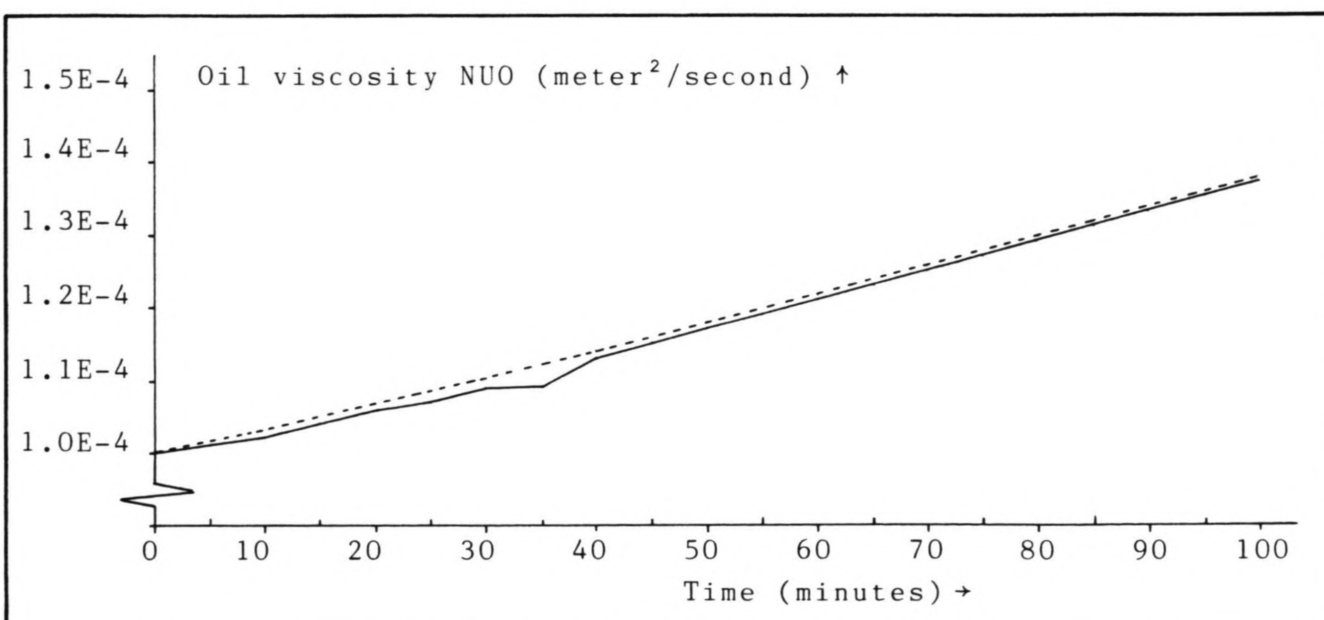


Figure 7.10. The SMOSS and MS4 curve of NUO.

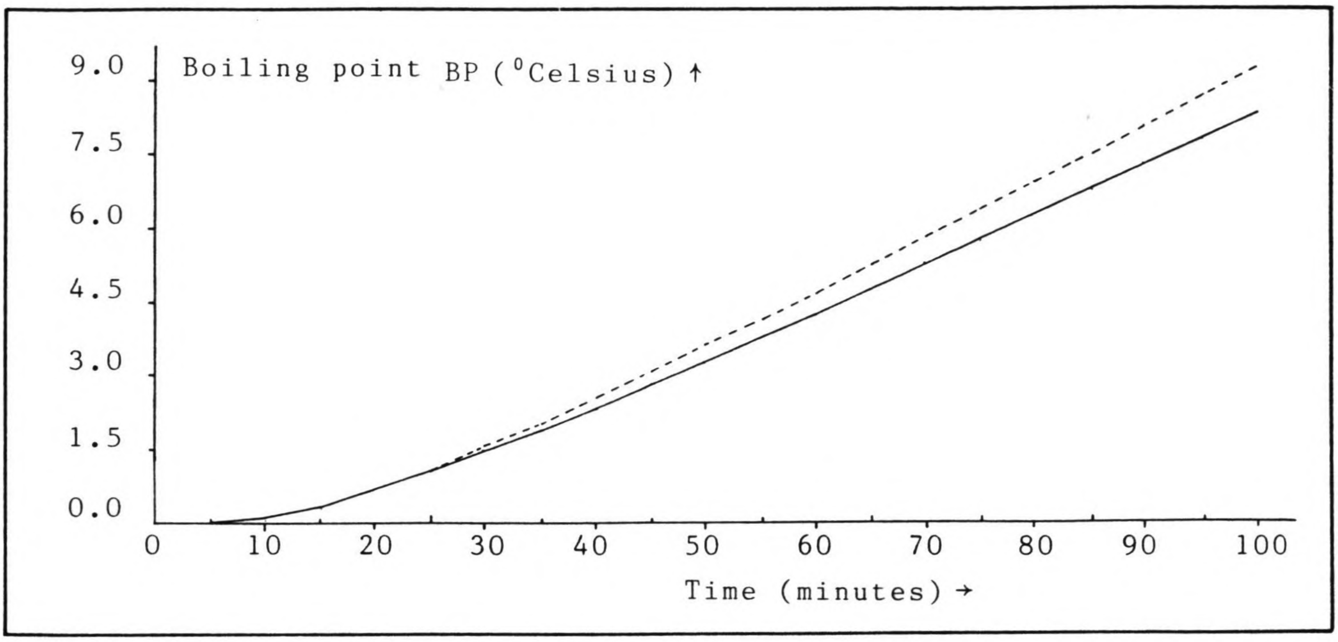


Figure 7.11. The SMOSS and MS4 curve of BP.

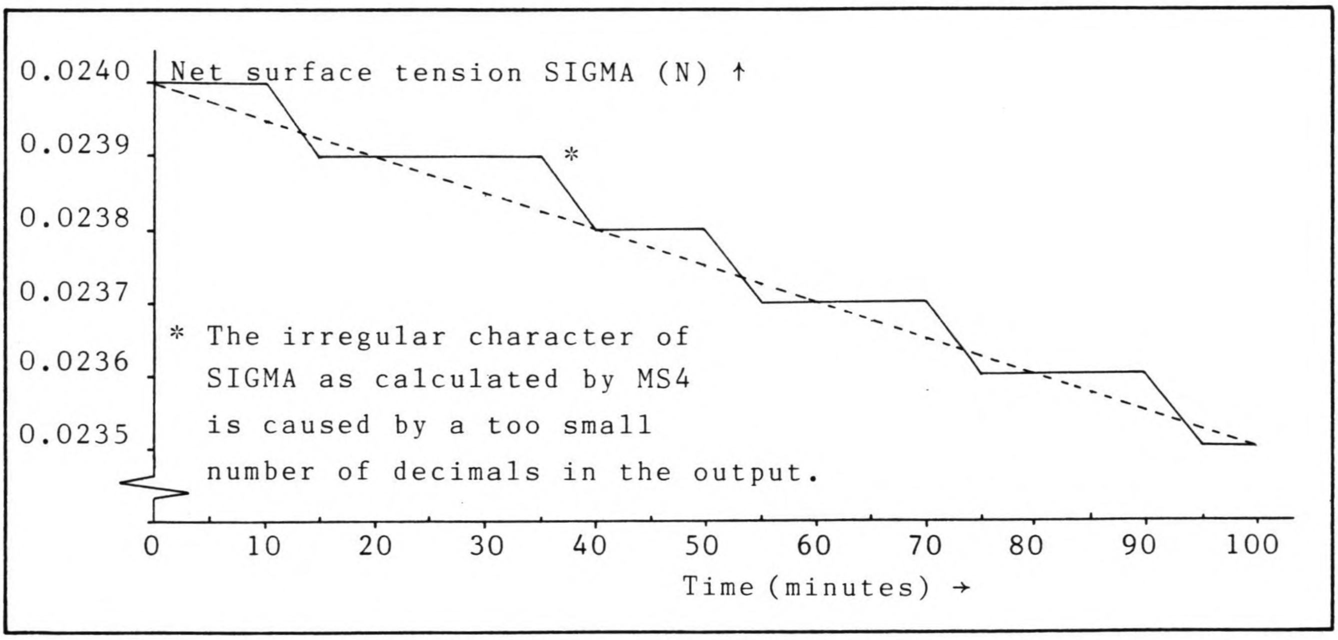


Figure 7.12. The SMOSS and MS4 curve of SIGMA.

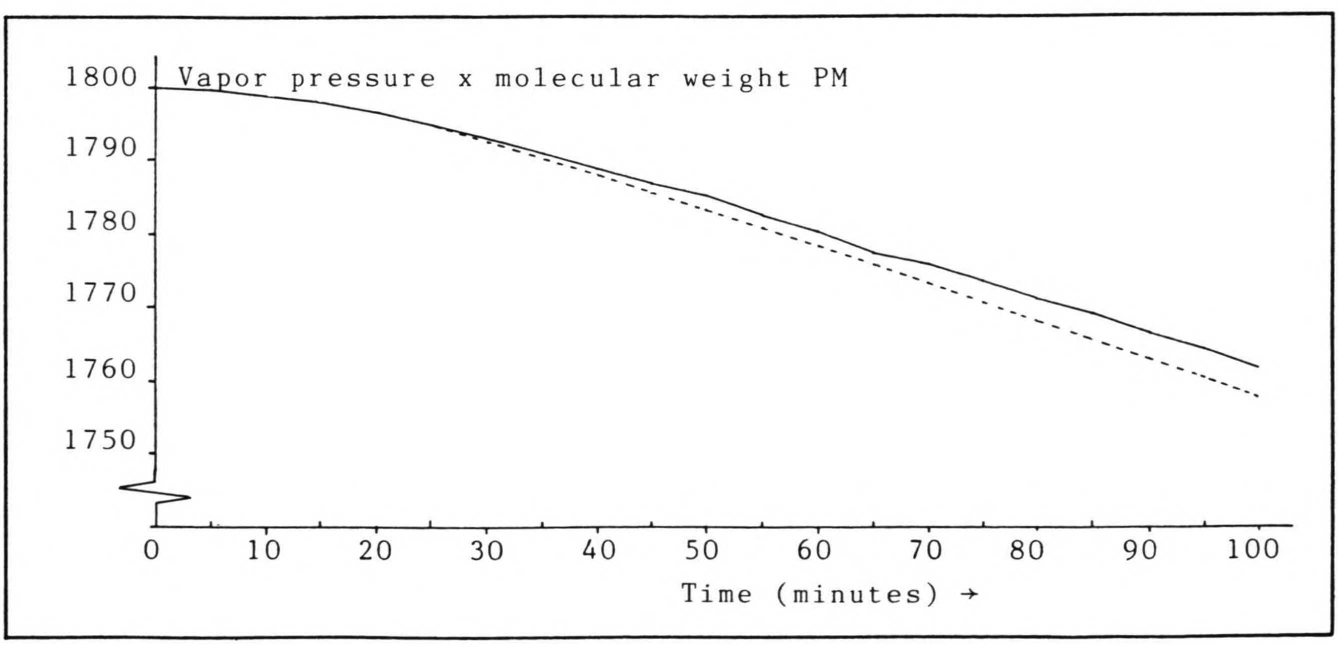


Figure 7.13. The SMOSS and MS4 curve of PM.

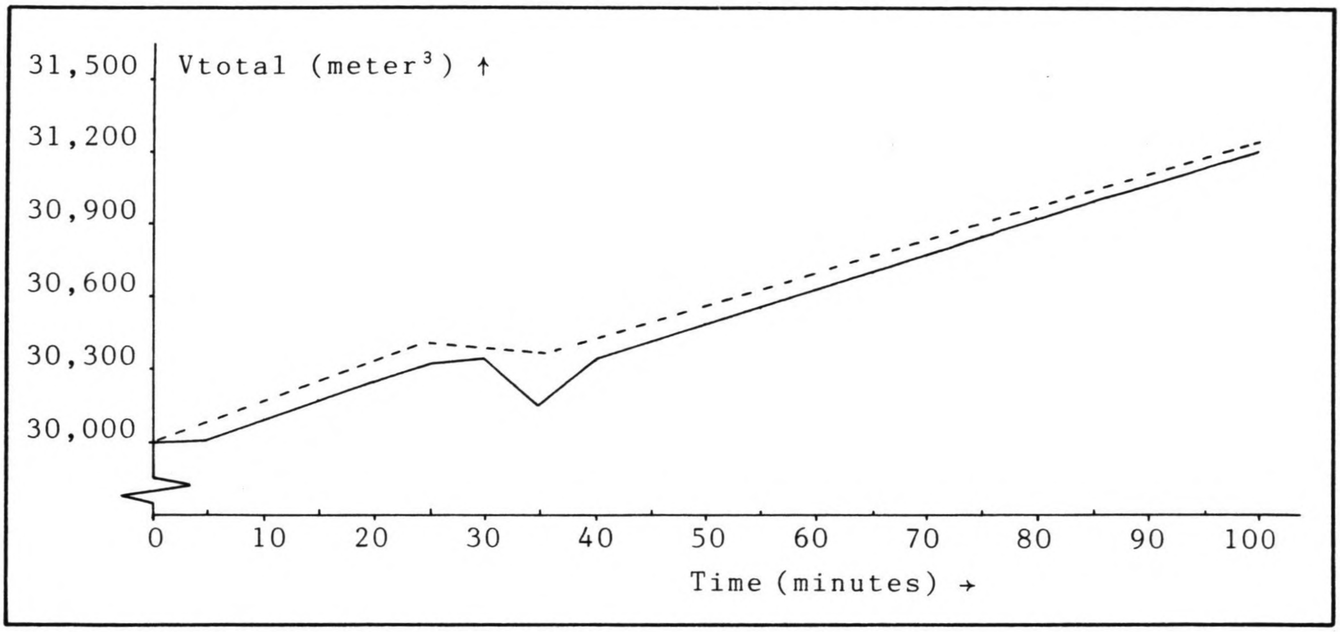


Figure 7.14. The SMOSS and MS4 curve of VTOTAL.

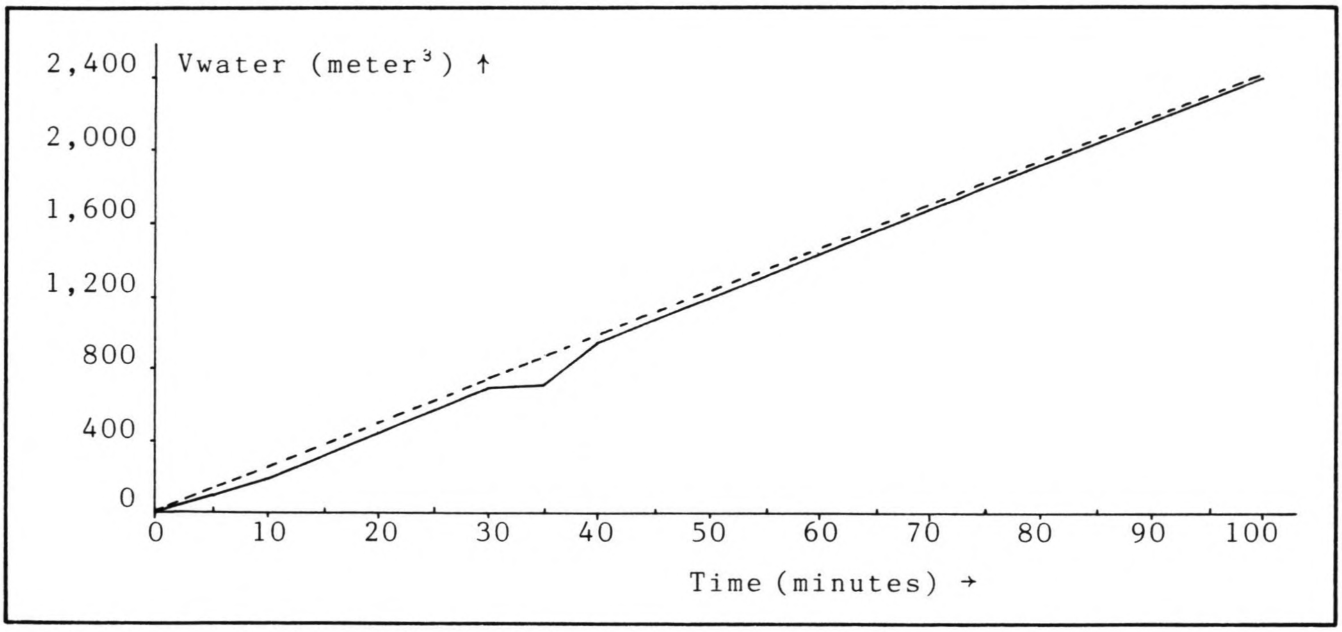


Figure 7.15. The SMOSS and MS4 curve of VWATER.

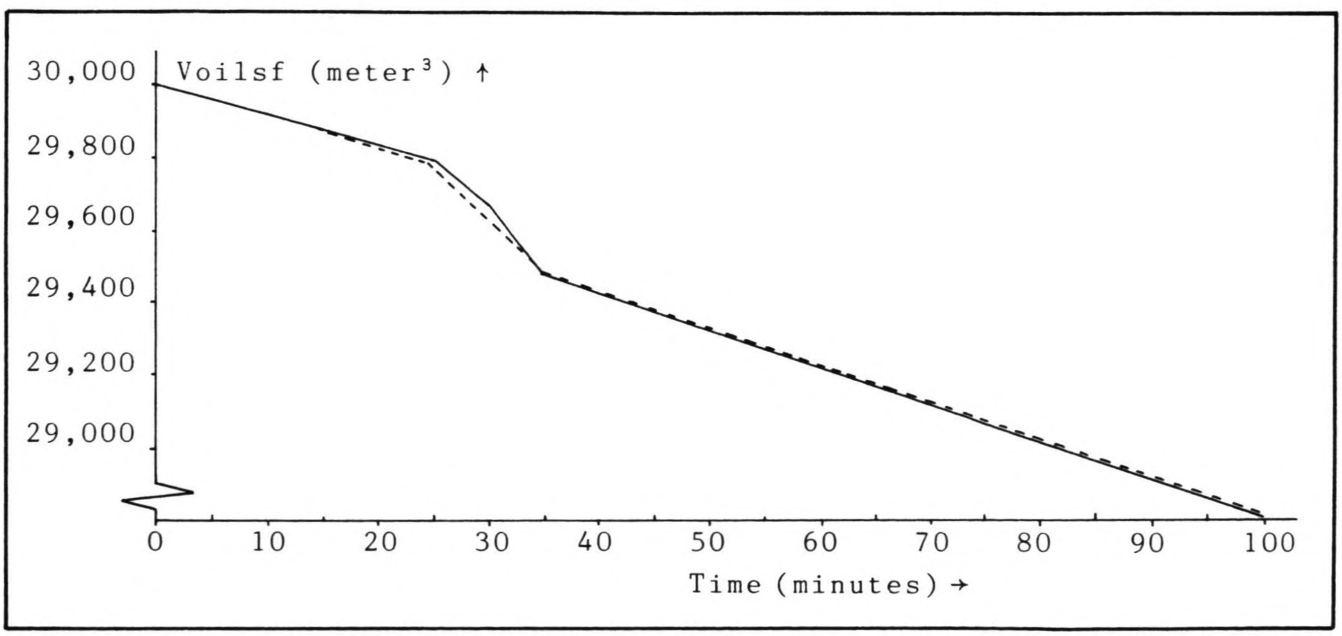


Figure 7.16. The SMOSS and MS4 curve of VOILSF.

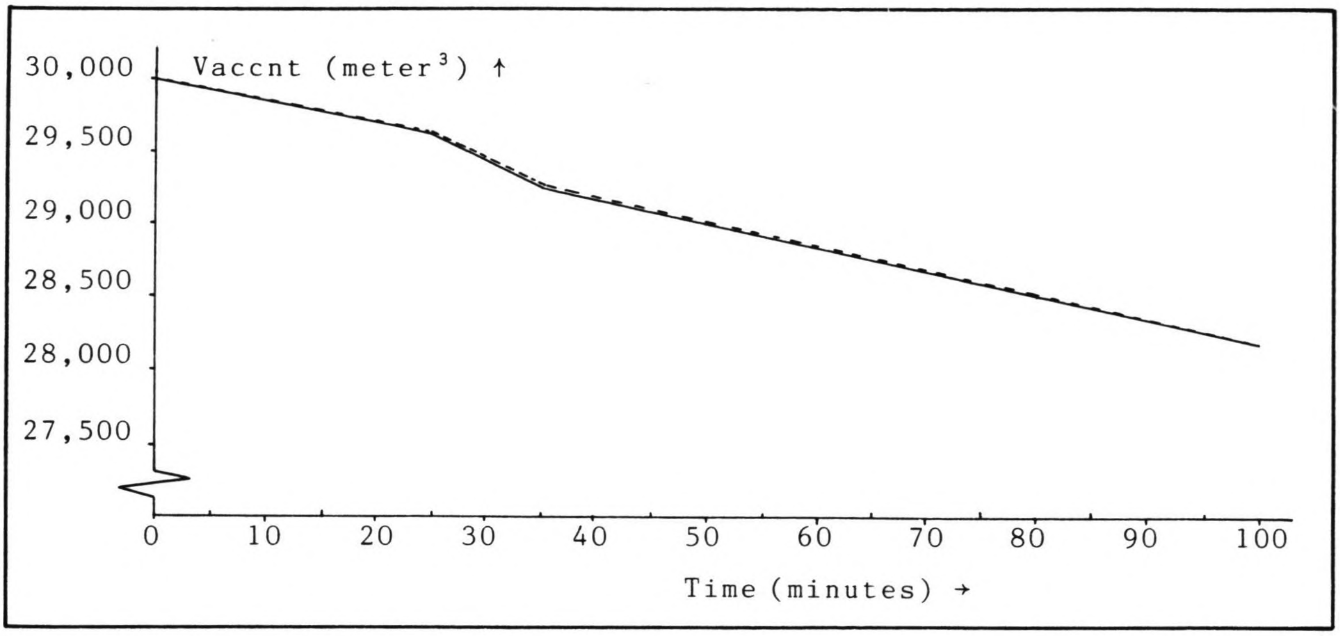


Figure 7.17. The SMOSS and MS4 curve of VACCNT.

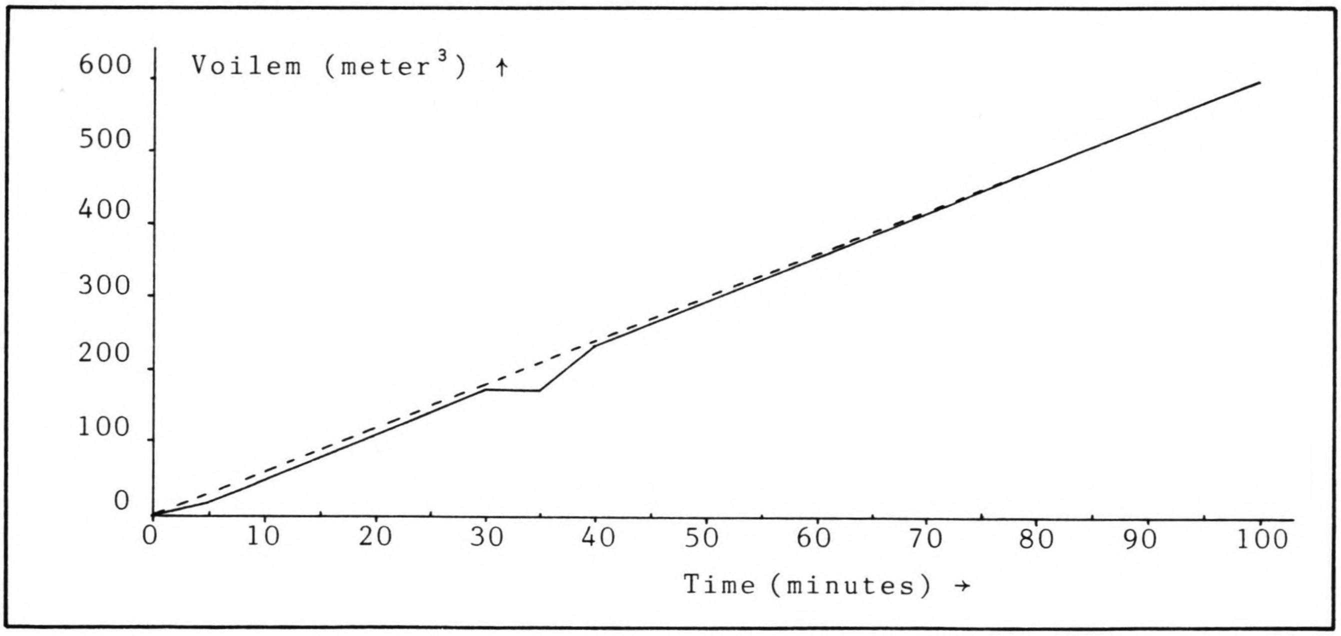


Figure 7.18. The SMOSS and MS4 curve of VOILEM.

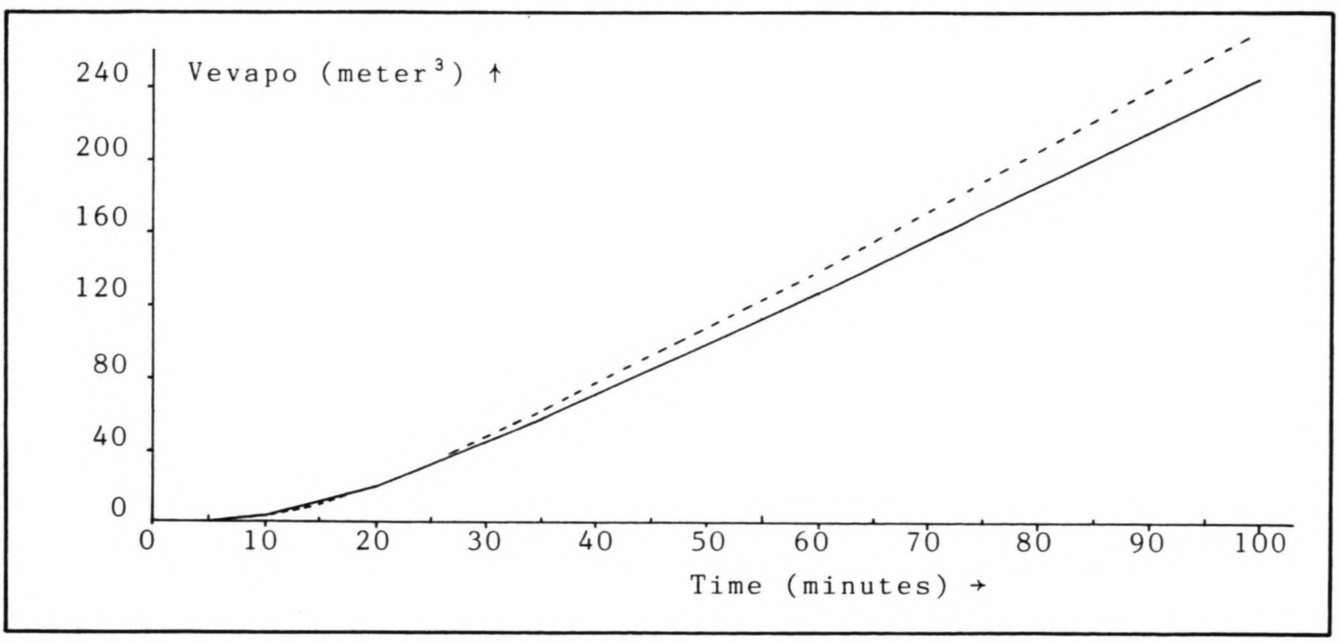


Figure 7.19. The SMOSS and MS4 curve of VEVAP.

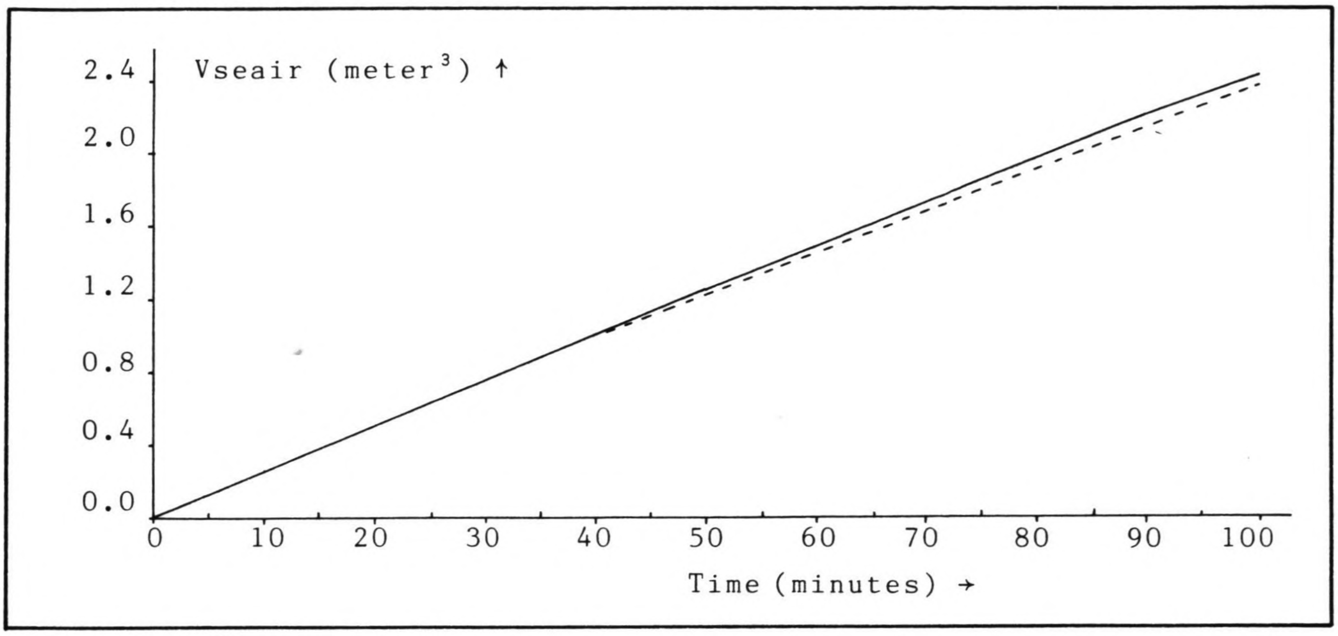


Figure 7.20. The SMOSS and MS4 curve of VSEAA.

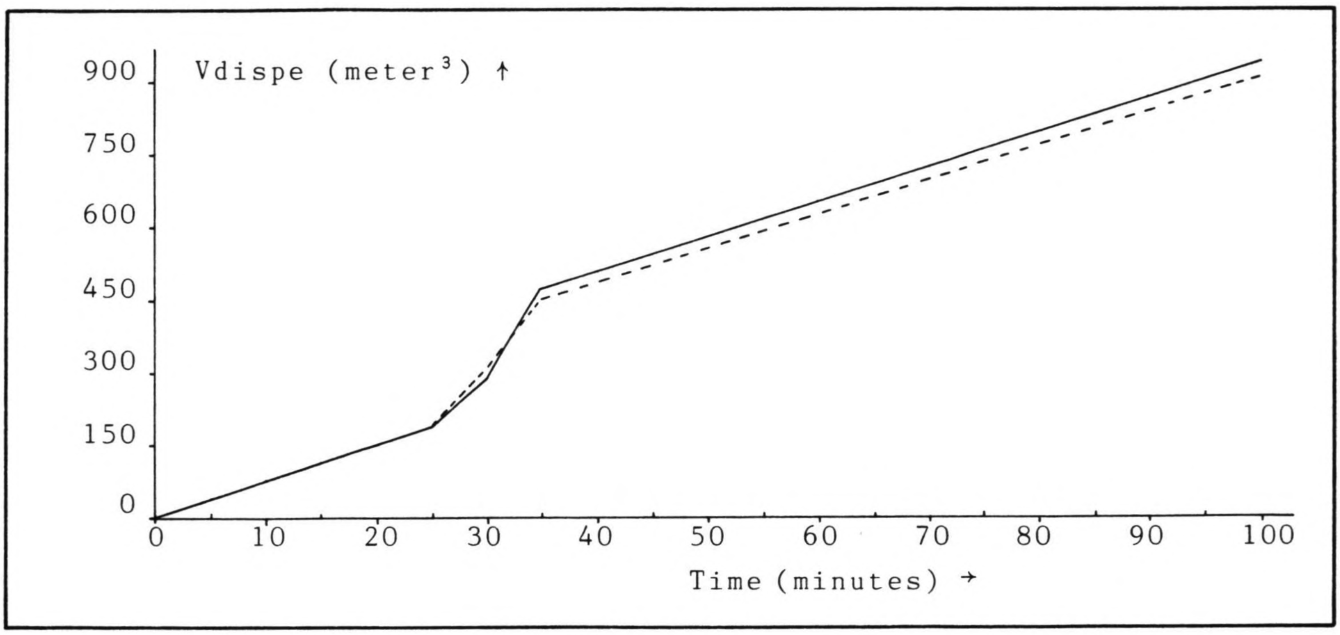


Figure 7.21. The SMOSS and MS4 curve of VDSP.

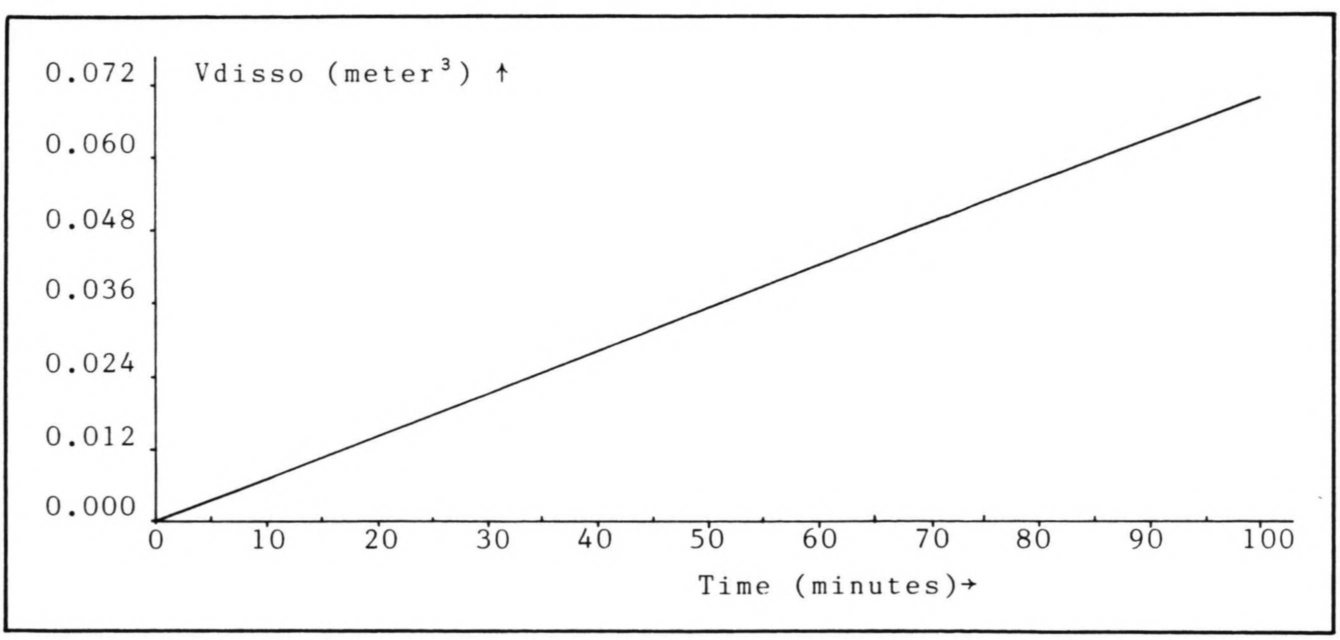


Figure 7.22. The SMOSS and MS4 curve of VDSS.

7.3. Second test run: 100 hours

With regard to SMOSS results reference is made to:

- SMOSS
  A Simulation Model for Oil Slicks at Sea
  Appendix E
  Paragraph E.4. 360000 seconds run (4 days)

With regard to MS4 results reference is made to:

- Appendix J (seperate volume)

Two curves are plotted in each graph.
The SMOSS curve: - - - - - - - - - - - - - - - - -
The MS4 curve : ——————————————

Because MS4 uses an essentially different interpolation program for wave, wind, tidal and current data, the data which were input in the data accumulation phase did not lead - after interpolation - to the same initial data as used by SMOSS. In order to obtain the same initial data the files have manually been corrected.

The vertical scales of the following graphs are mostly linear scales. In some cases however a logarithmic scale has been used.

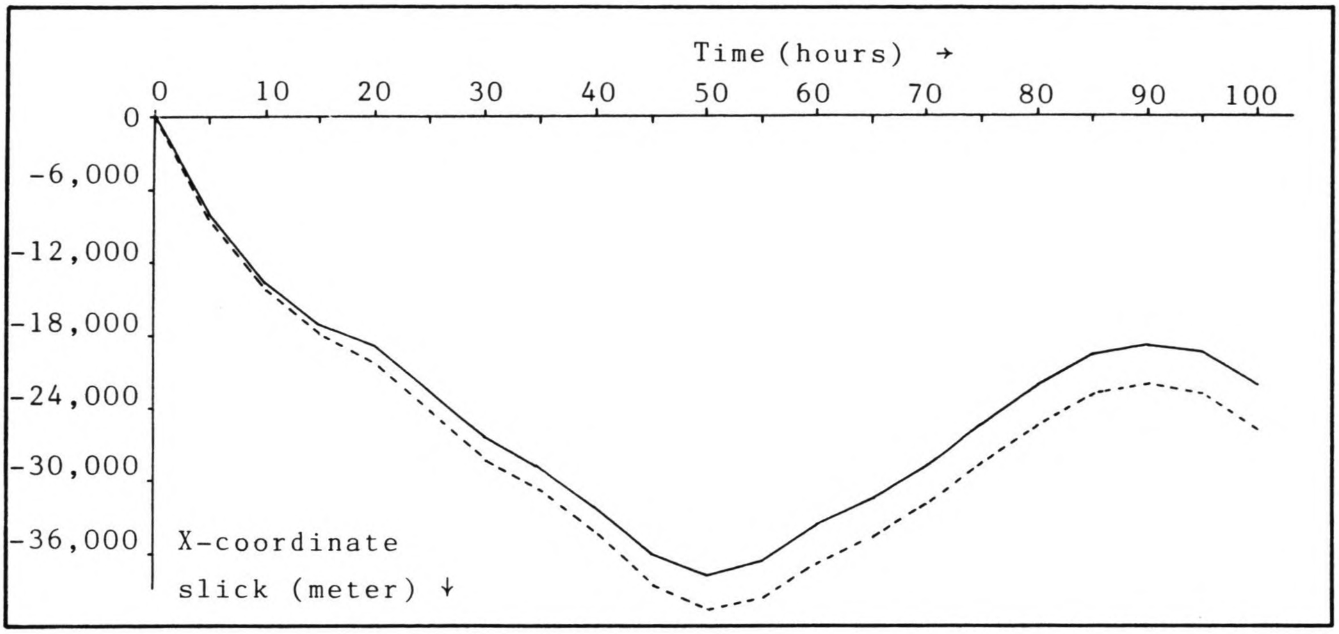


Figure 7.23. The SMOSS and MS4 curve of X.

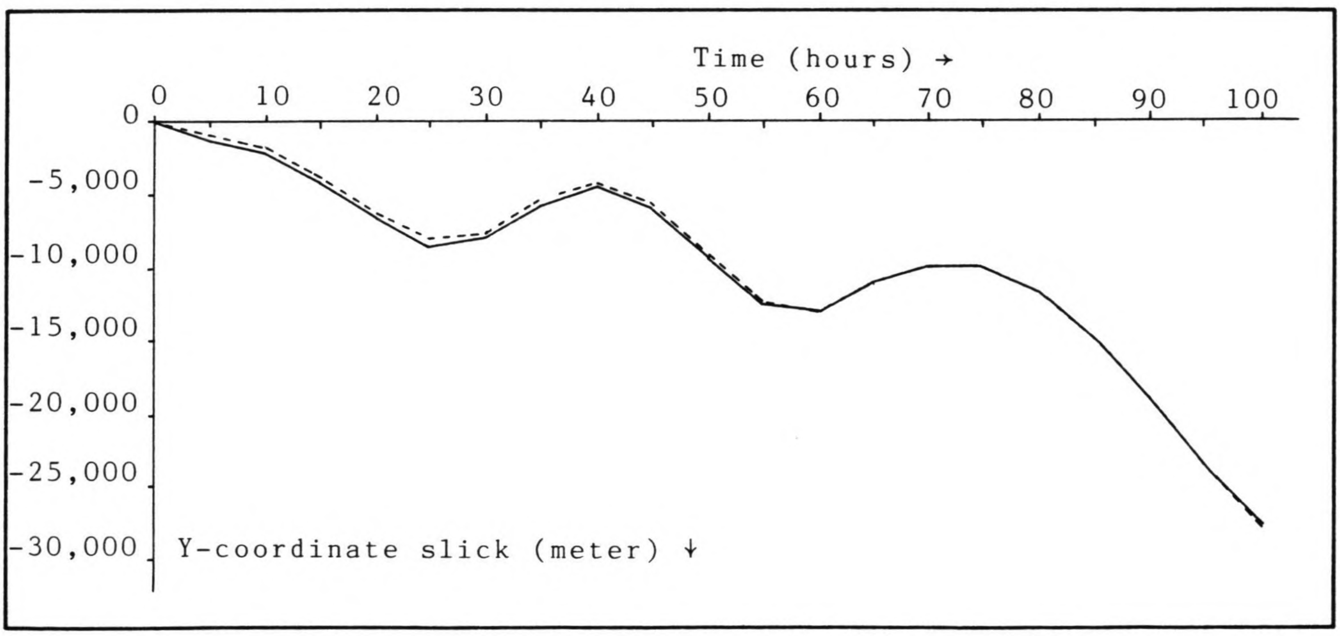


Figure 7.24. The SMOSS and MS4 curve of Y.

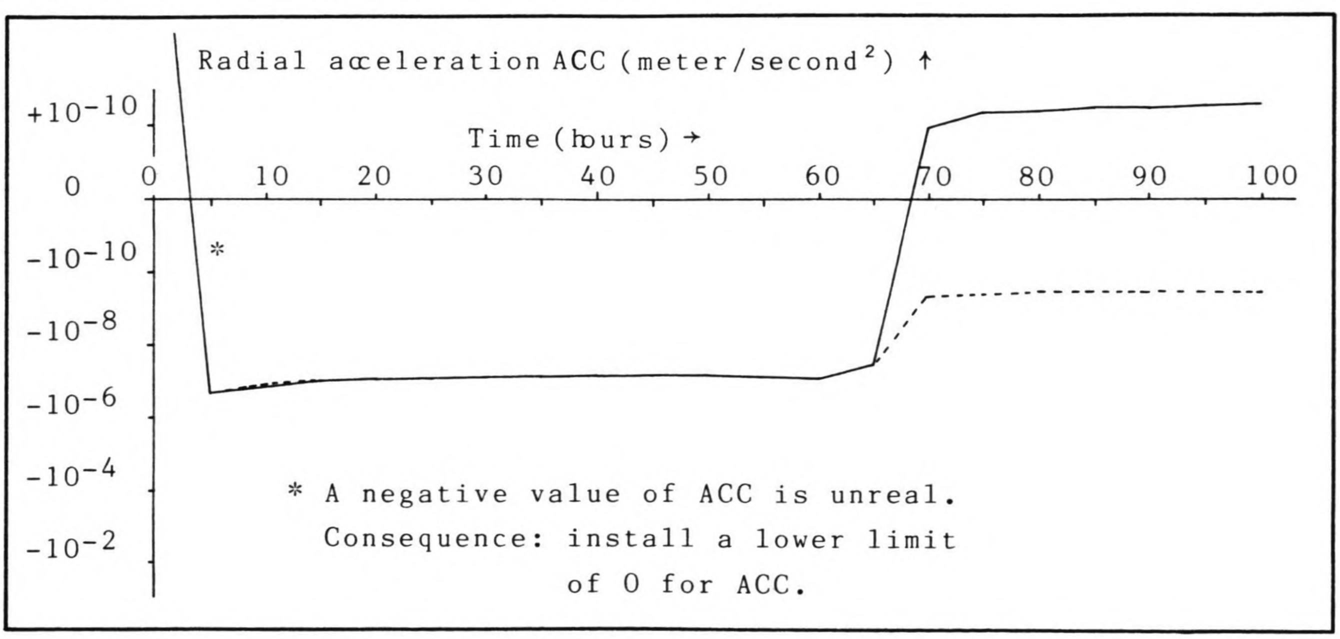


Figure 7.25. The SMOSS and MS4 curve of ACC.

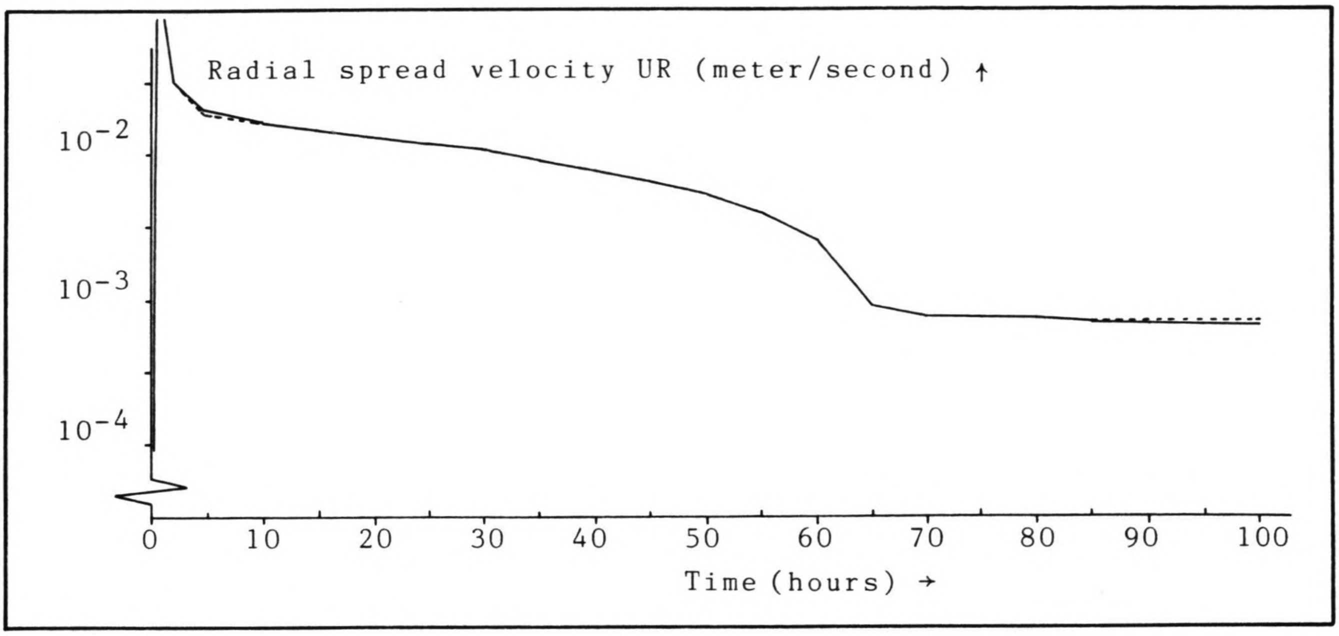


Figure 7.26. The SMOSS and MS4 curve of UR.

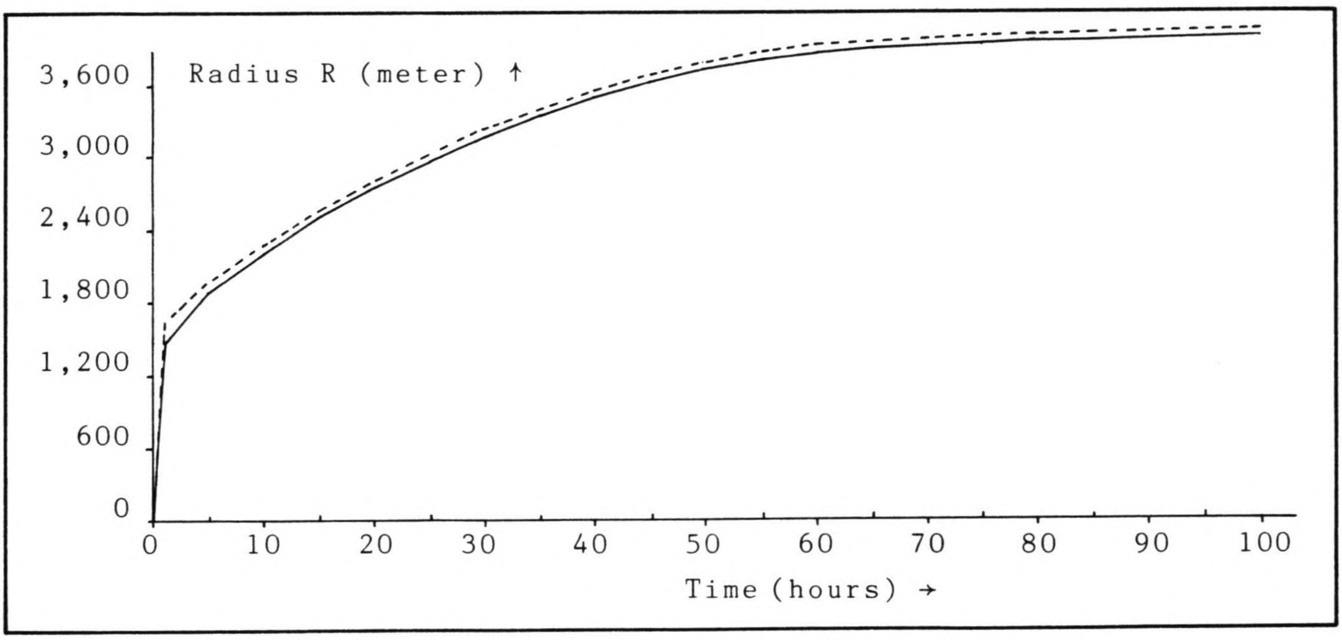


Figure 7.27. The SMOSS and MS4 curve of R.

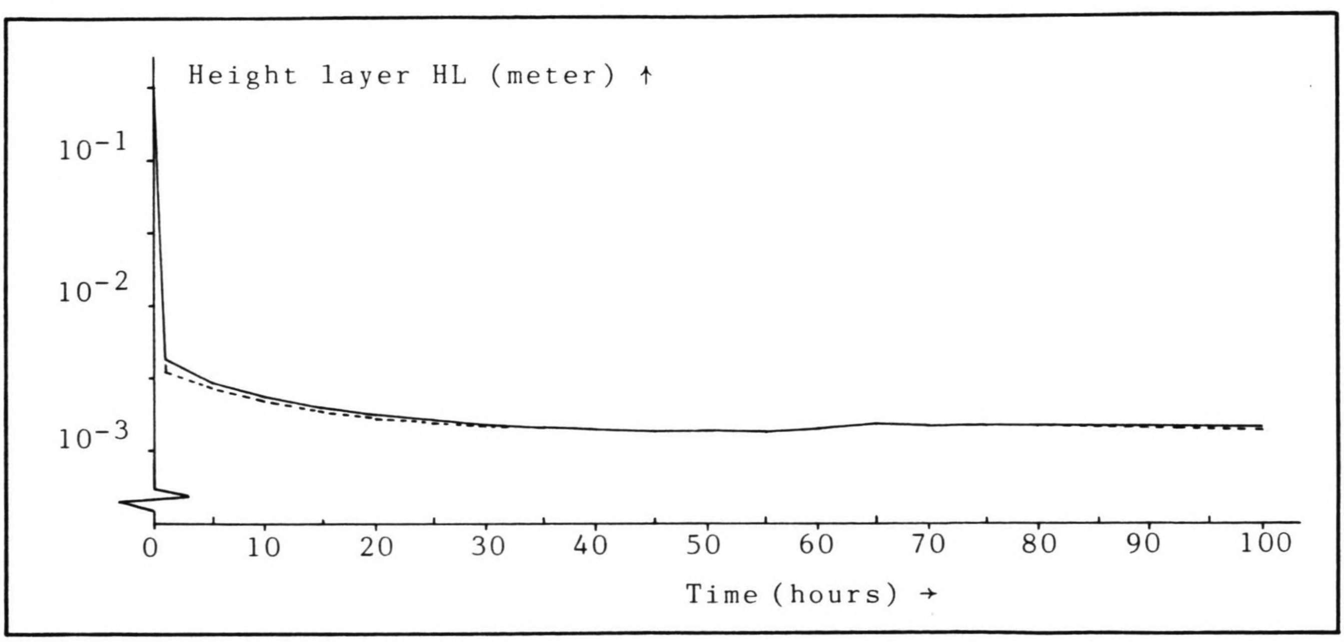


Figure 7.28. The SMOSS and MS4 curve of HL.

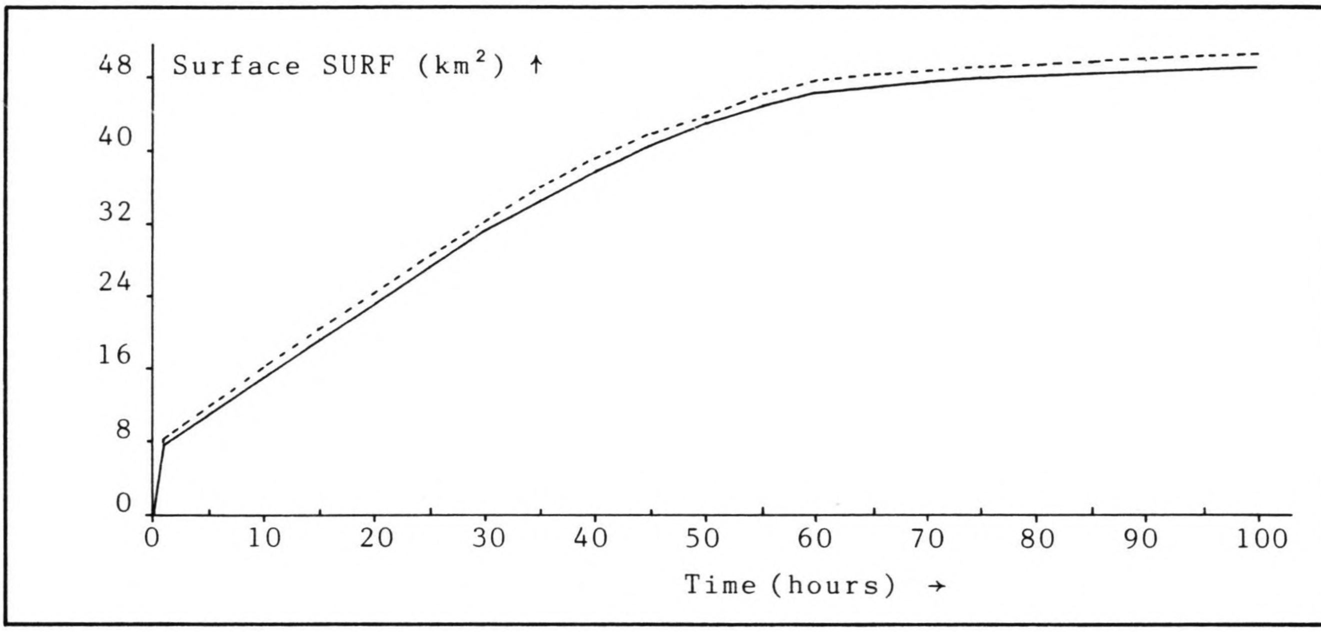


Figure 7.29. The SMOSS and MS4 curve of SURF.

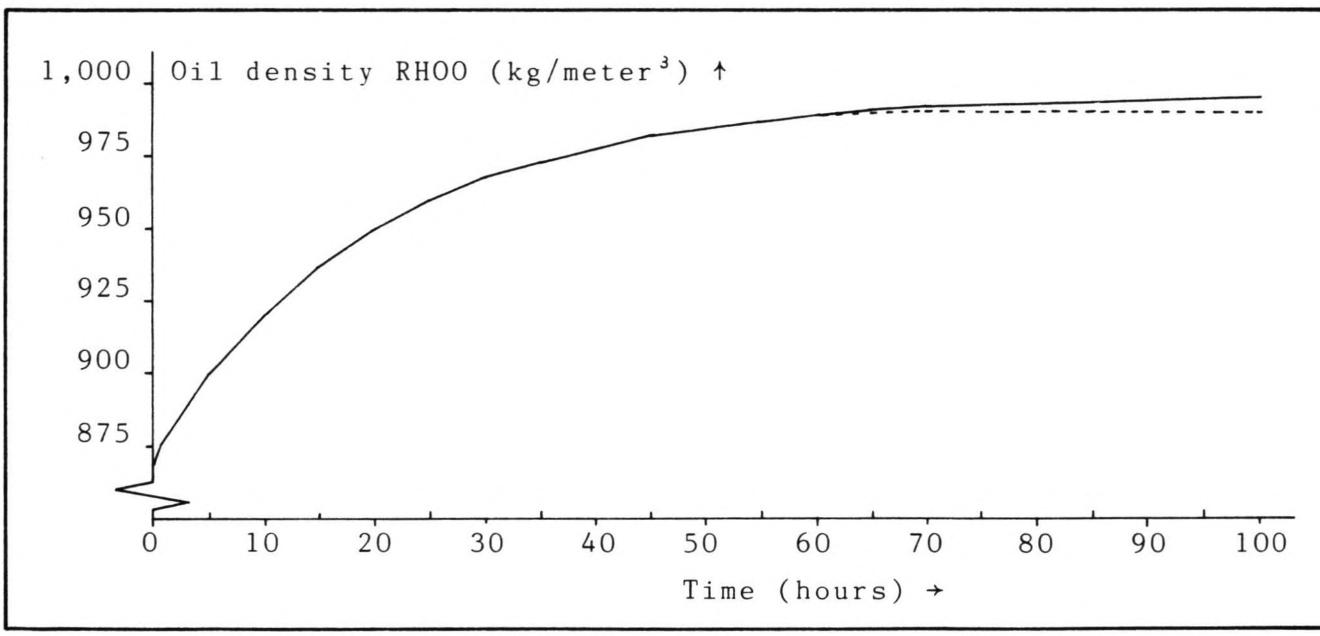


Figure 7.30. The SMOSS and MS4 curve of RHOO.

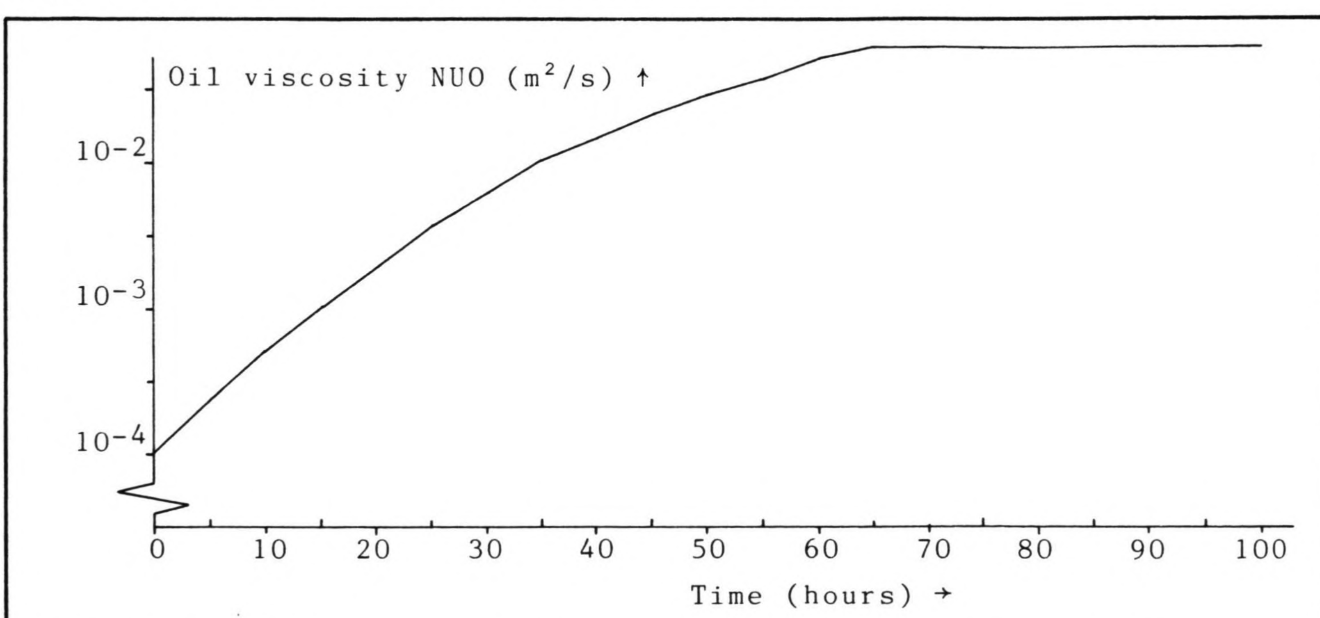


Figure 7.31. The SMOSS and MS4 curve of NUO.

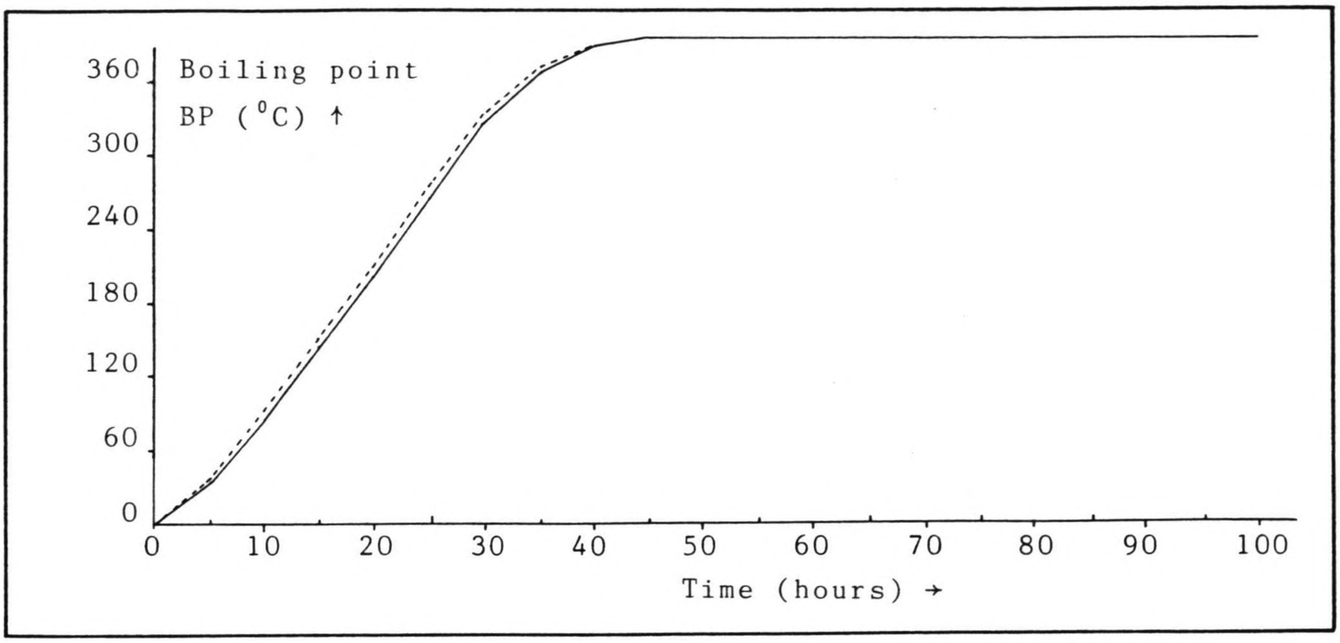


Figure 7.32. The SMOSS and MS4 curve of BP.

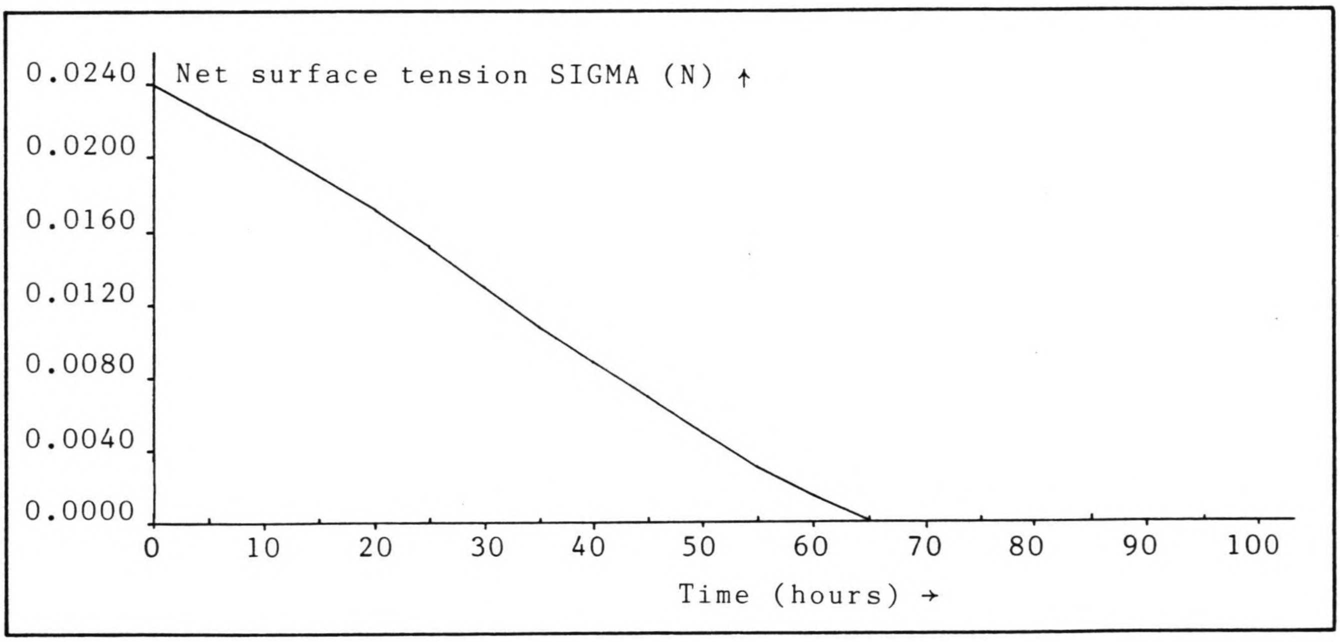


Figure 7.33. The SMOSS and MS4 curve of SIGMA.

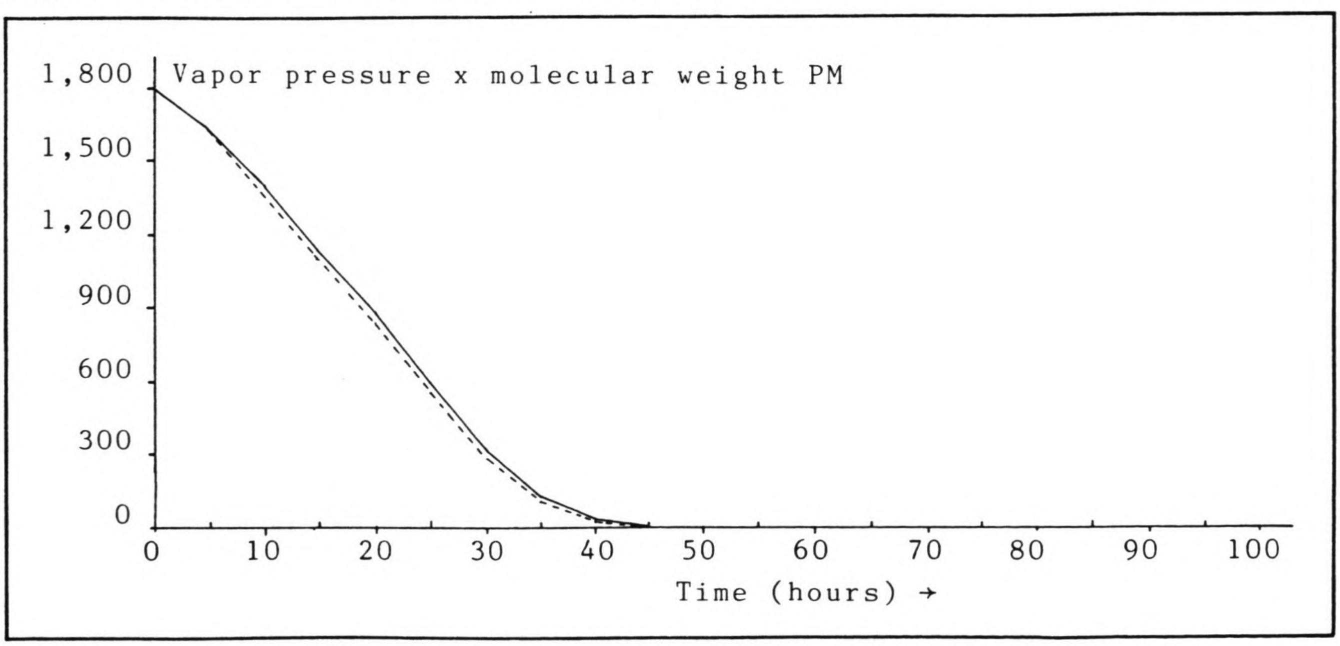


Figure 7.34. The SMOSS and MS4 curve of PM.

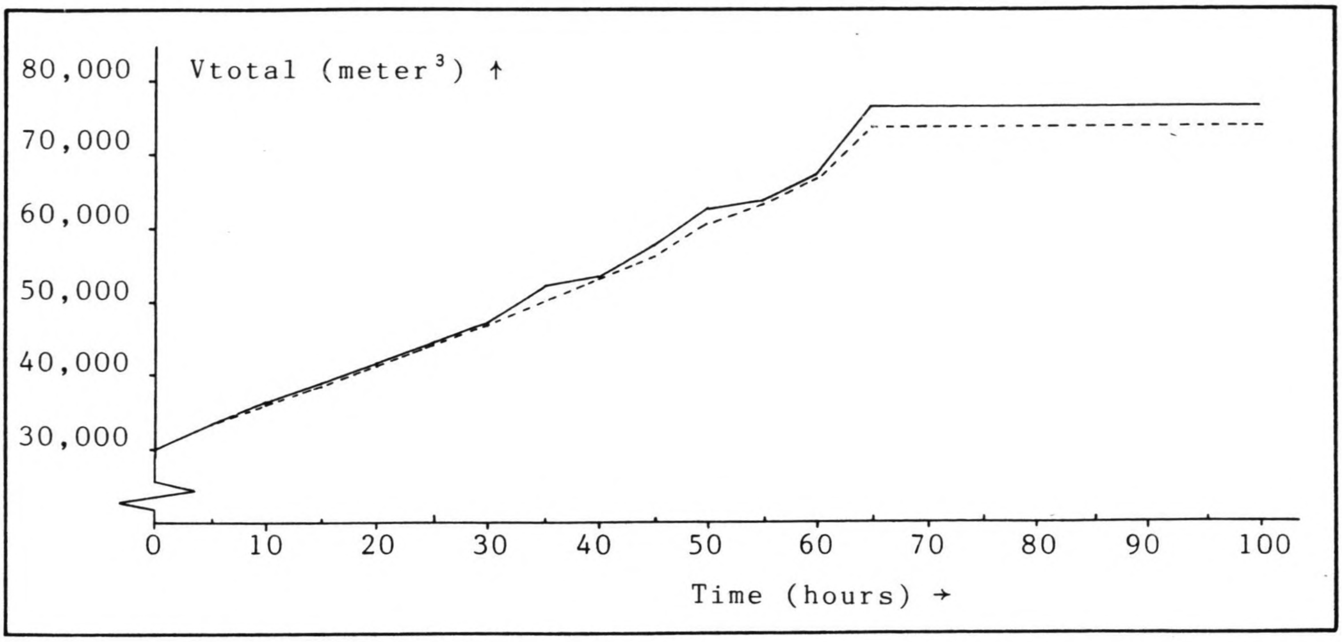


Figure 7.35. The SMOSS and MS4 curve of VTOTAL.

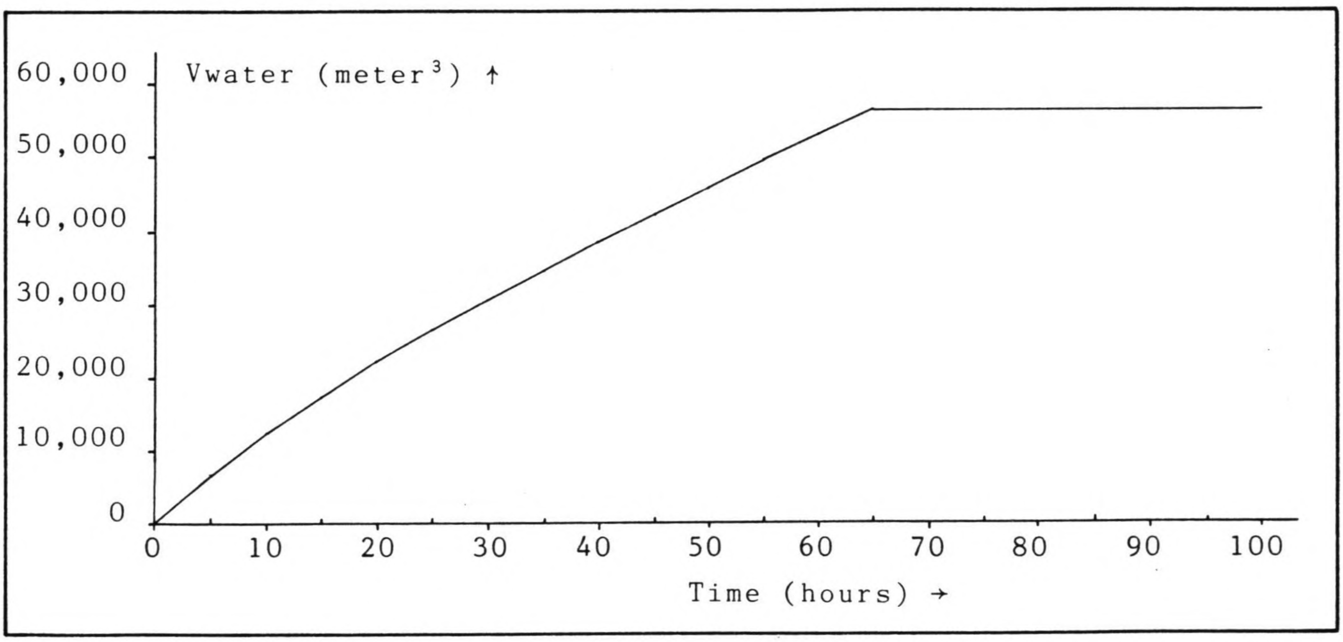


Figure 7.36. The SMOSS and MS4 curve of VWATER.

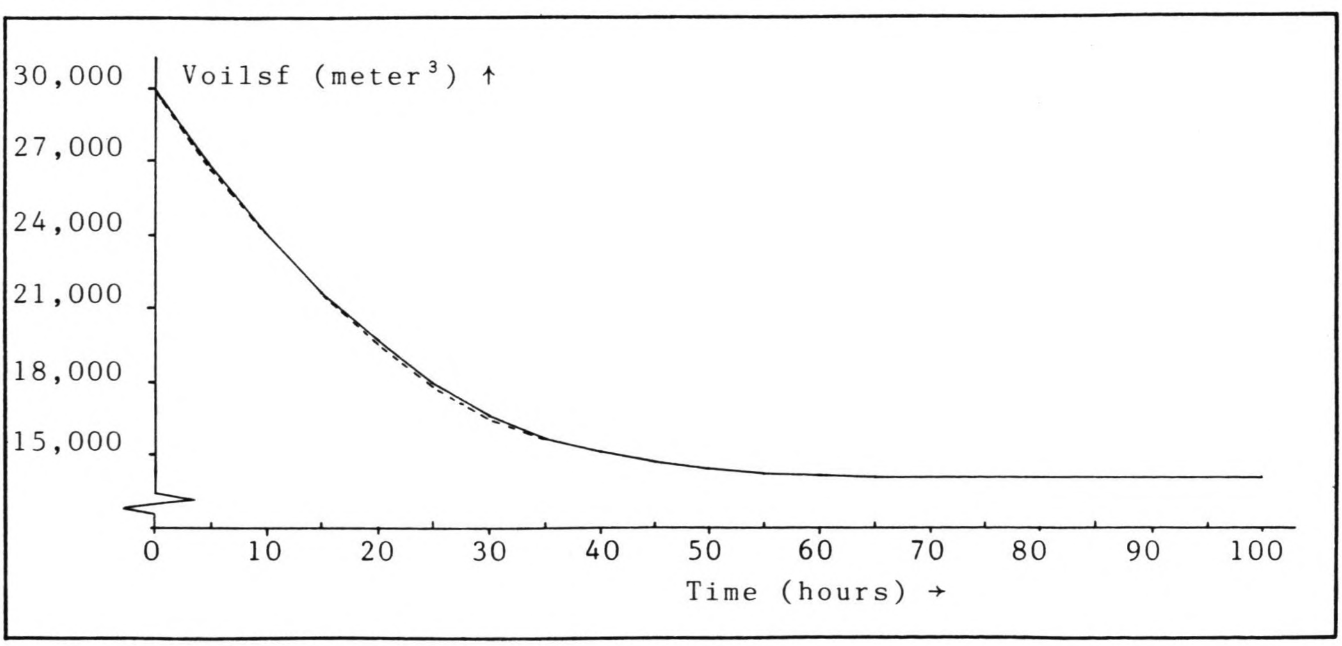


Figure 7.37. The SMOSS and MS4 curve of VOILSF.

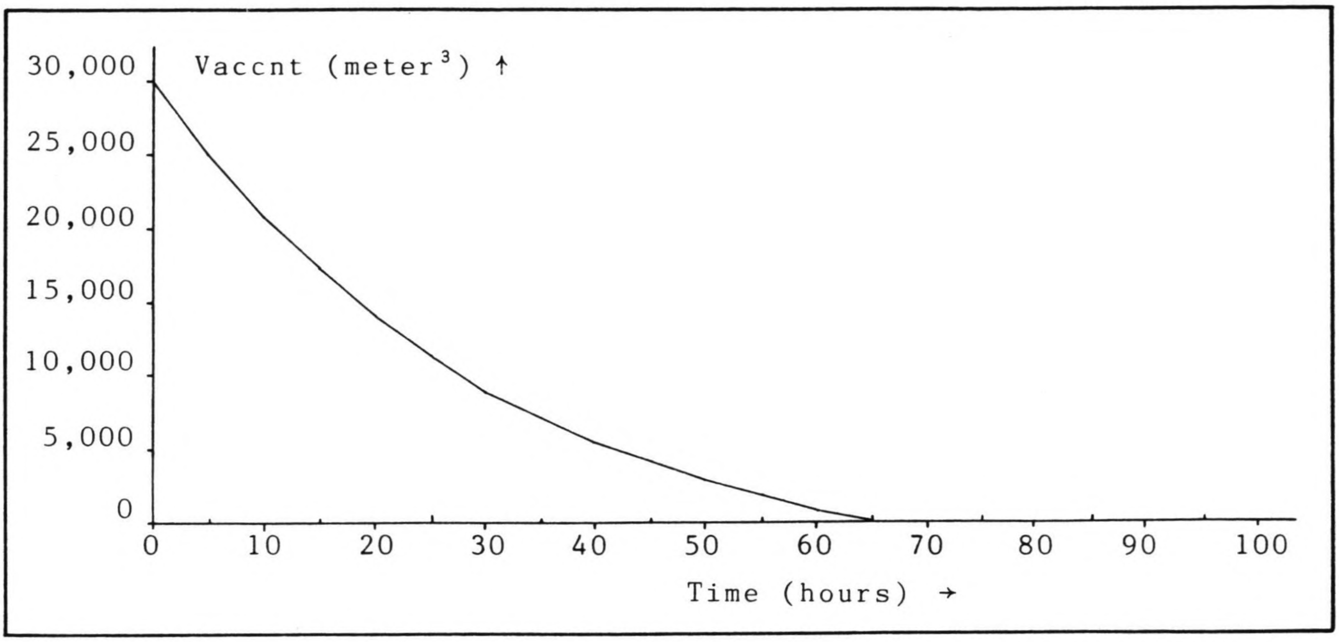


Figure 7.38. The SMOSS and MS4 curve of VACCNT.

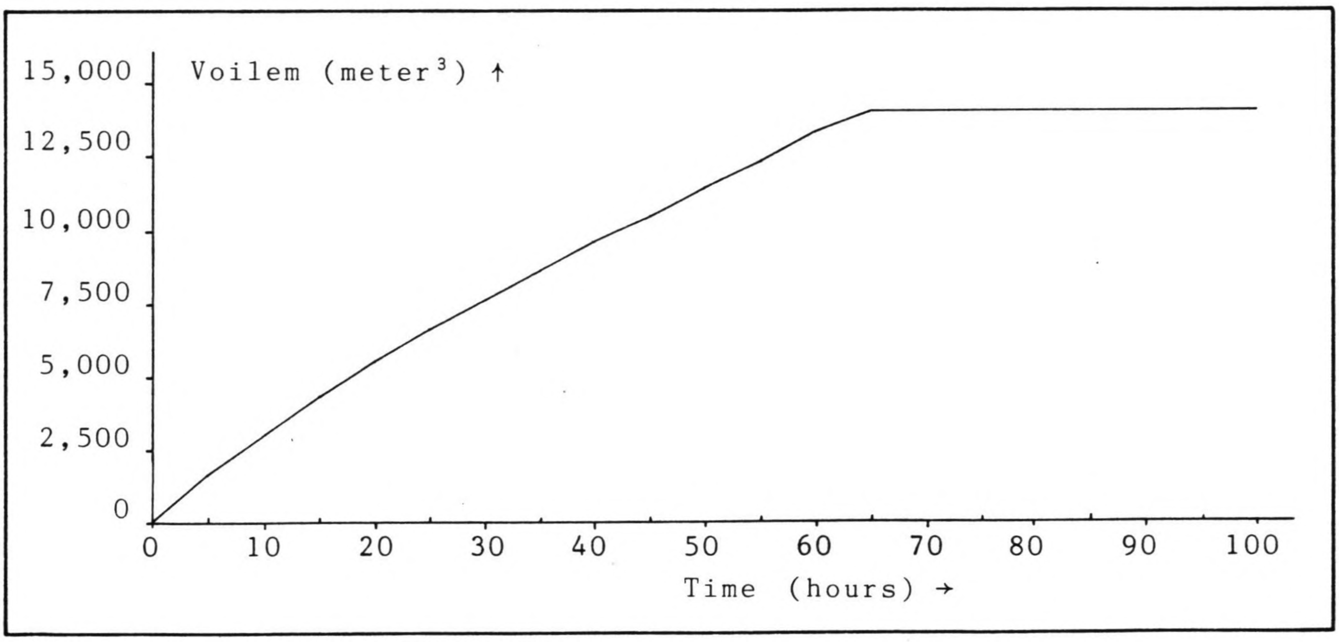


Figure 7.39. The SMOSS and MS4 curve of VOILEM.

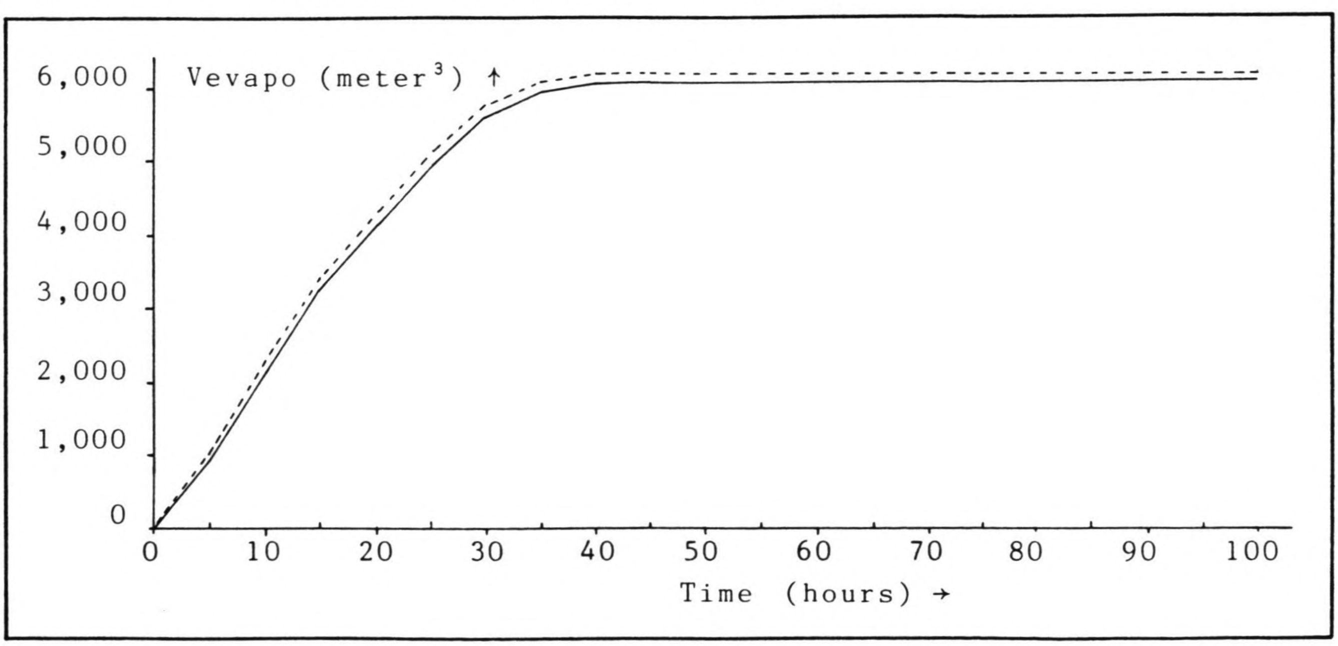


Figure 7.40. The SMOSS and MS4 curve of VEVAP.

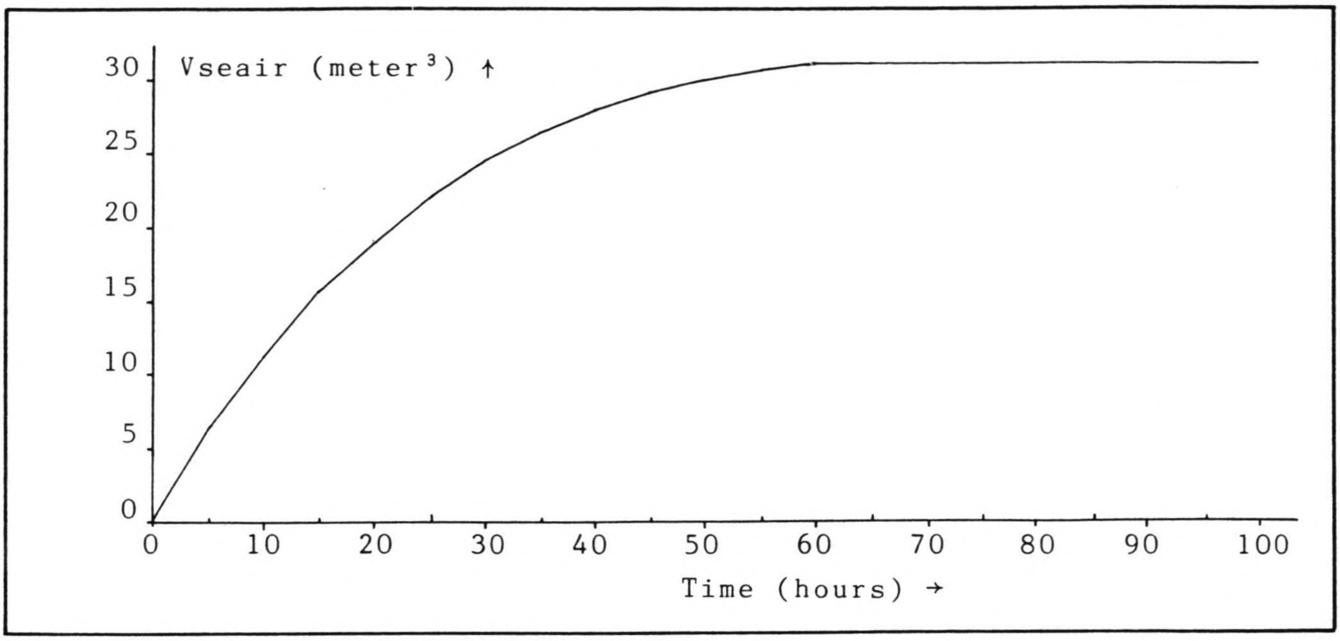


Figure 7.41. The SMOSS and MS4 curve of VSEAA.

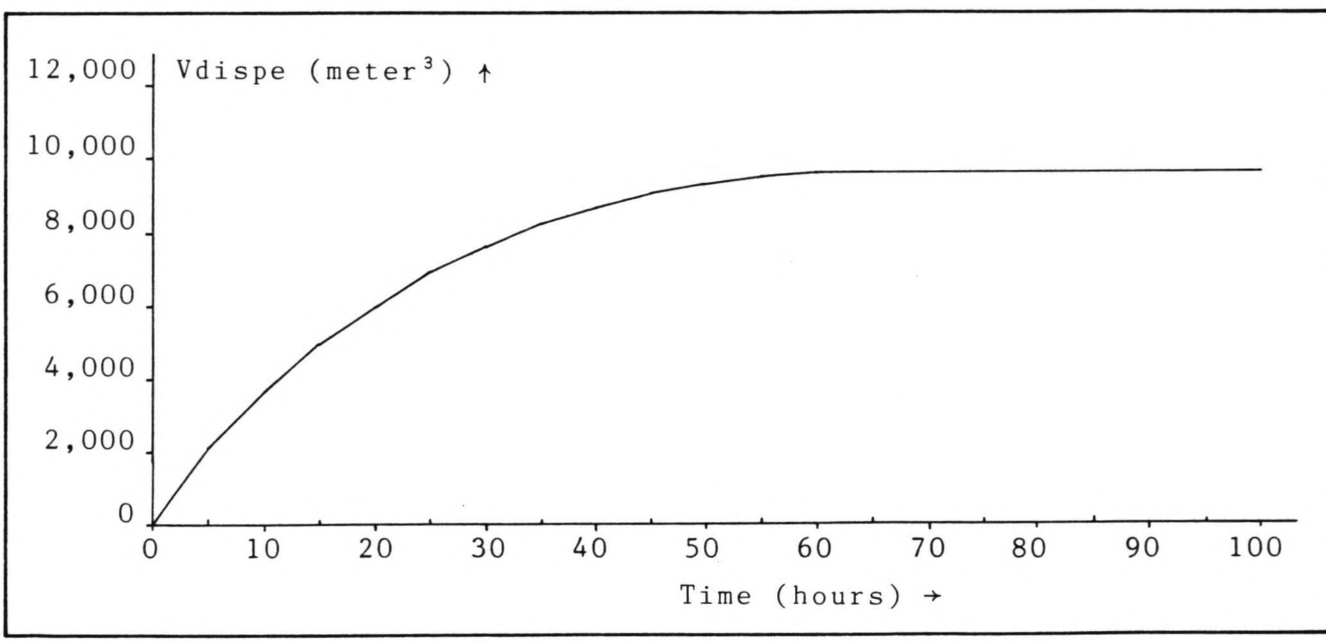


Figure 7.42. The SMOSS and MS4 curve of VDSP.

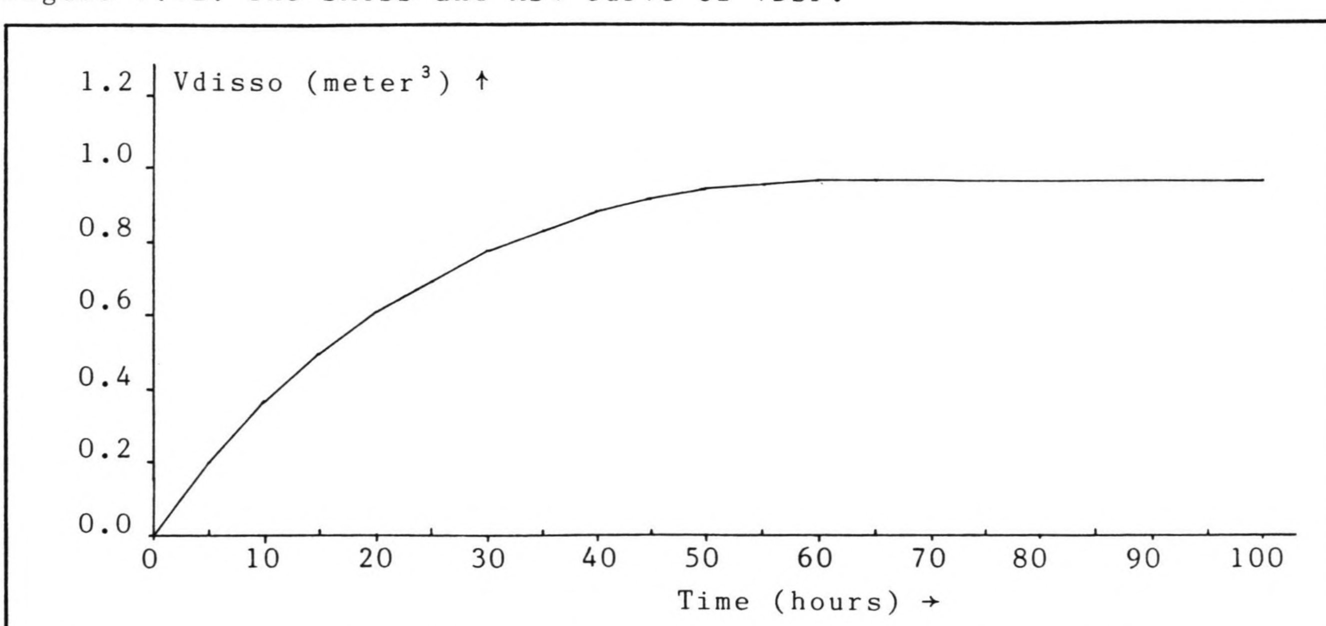


Figure 7.43. The SMOSS and MS4 curve of VDSS.

7.4. Accuracy

In general:

Accuracy: - freedom from mistake
- correctness, precision

With regard to the first meaning of the word accuracy - freedom from mistake - it can be said that this aspect of accuracy is usually very evident.
At the start of program testing very different results have been found due to incorrectly used parameters and programming errors, for instance: multiplication (*) instead of exponentiation (**).
All mistakes of this kind have been dealt with in the beginning of program testing and with regard to the first meaning of the word accuracy the model can be claimed to be as accurate as SMOSS.

The second meaning of the word accuracy - correctness, precision - is rather more complicated.
Here accuracy can be interpreted as a measure of all added errors. There are several types of errors.

Types of errors:

1. Model errors originate from the fact that a mathematical problem is just a simplified description of the physical problem. However, in the comparison between MS4 and SMOSS these errors do not exist because both are identical mathematical descriptions.

2. Calculation errors can be surpressed by executing checking-calculations. However, these errors do not apply as well because a computer in principle does not make these kind of mistakes; only people do!

3. Programming errors result in executing - by themselves correct - algorithms leading to incorrect results. These errors have already been discussed on the preceding page.

4. Rounding errors originate from calculations being executed with a fixed number of decimals. The limitation of the computer causes rounding of the numbers. This error equals a half unit of the last decimal at the most. Rounding errors can thus be reduced by executing calculations with numbers with more decimals. SMOSS used DOUBLE PRECISION (REAL*8) constants only. MS4 uses REAL*4 constants, so larger rounding errors will result but less memory space is needed. Insight in the size of these rounding errors will be obtained by running MS4 also with DOUBLE PRECISION constants. See figure 7.54..

5. Initial errors result from the initial data on which a calculation is based. In case of measured values there is already a certain amount of inaccuracy in these measurements. In case the initial data are based on predictions the initial error is rather self-evident. The results have at the most the same degree of accuracy as the initial data. Initial errors can only be reduced by making the initial data more accurate. With regard to the comparison between SMOSS and MS4 - the program testing - the remark must be made that to achieve a correct comparison, all initial data should be the same so no errors will result on these grounds. This condition has been fulfilled for both test runs.

6. Truncation errors are errors corresponding to the fact that a (finite or infinite) sequence of computational steps necessary to produce an exact result is truncated prematurely after a certain number of steps. These errors depend on the computational method and have to be discussed individually with each method. This type of error is especially important for the integration routines as used in the model. SMOSS used the CSMP sophisticated integration method STIFF, an integration method used for handling stiff equations. MS4 uses - see chapter 5 - the very simple trapezoidal rule. This enormous difference in choice of integration methods must of course lead to different truncation errors.

In summary, when comparing SMOSS with MS4 there are two possible types of errors: rounding and truncation errors. Of these two only the last type of error is expected to be really determining for possible differences in results.

<u>With regard to test run results:</u>

- the graphs of paragraph 7.2. and 7.3. give an excellent view of the achieved accuracy.

- Figures 7.44 and 7.45 list the relative errors for both test runs followed by an analysis.

- the graph showing accuracy versus integration step is placed in the paragraph dealing with efficiency.

100 minutes test run:

Variable	X	Y	ACC	UR	R	HL	SURF
Rel. error $^{\alpha}$	5.7%	3.3%	18.3%	6.1%	6.1%	0.0%	12.2%
Variable	RHOO $^{\beta}$	NUO	BP	SIGMA	PM	VTOTAL $^{\beta}$	VWATER
Rel. error	0.0%	0.0%	11.2%	0.0%	2.4%	1.5%	1.1%
Variable	VOILSF $^{\beta}$	VACCNT $^{\beta}$	VOILEM	VEVAP	VSEAA	VDSP	VDSS
Rel. error	0.0%	0.0%	1.1%	1.1%	1.5%	3.1%	0.0%

Figure 7.44. The relative error for most variables of 10 minutes run.

ad α Rel. error = $|\{(value_{MS4} - value_{SMOSS})/value_{MS4}\}| *100\% \Big|_{time\ =\ 10\ min.}$

ad β Please bear in mind that these relative errors are not very representative due to the high initial values.

Analysis:

Based on the relative error percentages of figure 7.44. it can be concluded that the model has a high degree of accuracy.
Note: this accuracy refers to the comparison between SMOSS and MS4, two theoretical models.

The relative error of ACC is rather large: 18.3%. The absolute error, however, is roughly: 1.5 E-7. This small absolute error plays no role at all in the simulation at this point of time. The importance of ACC lies in this case in the first half hour of the simulation. Furthermore little differences like this do not essentially influence other variables.

100 hours test run:

Variable	X	Y	ACC	UR	R	HL	SURF
Rel. error $^{\alpha}$	16.7%	0.5%	179.1%	6.2%	1.6%	3.4%	3.2%
Variable	RHOO $^{\beta}$	NUO	BP	SIGMA	PM	VTOTAL $^{\beta}$	VWATER
Rel. error	5.2%	0.3%	0.0%	0.0%	γ	0.4%	0.4%
Variable	VOILSF $^{\beta}$	VACCNT	VOILEM	VEVAP	VSEAA	VDSP	VDSS
Rel. error	0.4%	δ	0.4%	1.9%	0.5%	0.7%	0.4%

Figure 7.45. The relative error for most variables of the 100 hour run.

ad α: Rel. error = $\left|\{(value_{MS4} - value_{SMOSS})/value_{MS4}\}\right| *100\% \Big|_{time\ =\ 100\ hours}$

ad β: Please bear in mind that these relative errors are not very representative due to the high initial values.

ad γ: No percentage is given due to difference in formatting: 0.0000 versus 1.7 E-16.

ad δ: No percentage is given due to small error in MS4. VACCNT has final value of -2.9 which is physically impossible. A lower limit will be installed in MS4.

Analysis:

Based on the relative error percentages of figure 7.45. it can be concluded that the model has a high degree of accuracy. However, there seem to be two exceptions: X and ACC.

The relative error of ACC may be 179.1% but the absolute error is roughly 7.1 E-10. This small absolute error plays no role at all in the simulation at this point of time.

The importance of ACC lies in this case in the first thirty minutes of the simulation. ACC only influences the radial spread velocity UR and the slick radius R directly. The relative error percentages of these variables are 6.2% resp. 1.6%. Conclusion: this relative error percentage of ACC can easily be accepted.

The relative error percentage of X is 16.7%. The reasons for this notable error are as follows:

- The initial data of waves and winds. Although the files have been manipulated to match the SMOSS input these initial data differ slightly sometimes. However, considering the 0.5% relative error percentage of Y this indicates this must be a minor reason indeed.

- The integration routine combined with the somewhat oscillating character of X. The fact that the oscillating character brings the X-curve back to the 0-axis causes an unjustified high error percentage. A more clarifying point of view is by calculating the X-deviation per hour. The X-deviation is 37 meter/hour or roughly 1 centimeter/second. This is not much!

7.5. Computer run time

Computer run time is the time required for executing the simulation. This run time is important in view of future additions or changes. Also the ratio in run time between computing and input/output operations is of importance to justificate the possible procuring of a better processor (computing) or a faster Winchester disk (input/output operations).

The computer run time has in this case only been related to the actual simulation and not to the data accumulation and output presentation. Now the actual simulation has a core consisting of three blocks:

1. Initial block :
   - Transfer input data to internal storage from data files using the formatted sequential READ statement
   - Give the initial values to all internally used parameters
   - Create the new file PARAME.DAT which lists the internal parameters
   - Create five new files and keep the logical units that connect these files open for output from the process block

2. Process block :
   - Determine integration step size spreading
   - Determine integration step size aging
   - Compute transport variables
   - Compute spreading variables
   - Compute aging variables
   - Transfer output data to five files using formatted sequential WRITE statements

3. Terminal block:
   - Close all files

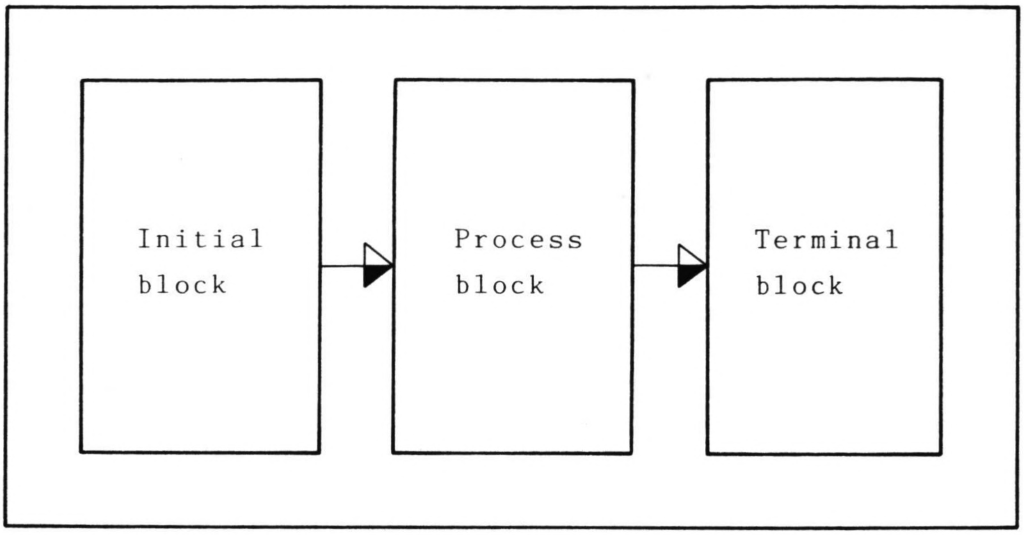


Figure 7.46. The core of the simulation.

To determine the computer run time the SECNDS function subprogram has been used, which returns the system time in seconds. In this way the SECNDS function can be used to perform elapsed-time computations. The value of SECNDS is accurate to the resolution of the system clock: 0.02 seconds for a 50-cycle clock.

Computer run time initial block:

$RT_{init} = 11.7 + 3.6 * n$ (seconds); n = number of simulated days

Computer run time terminal block:

$RT_{term} \approx 0$ (seconds)

Computer run time process block:

The following run times are valid for a single computation only; the proportion between the six elements for a single computation can differ substantially from that of a complete simulation.

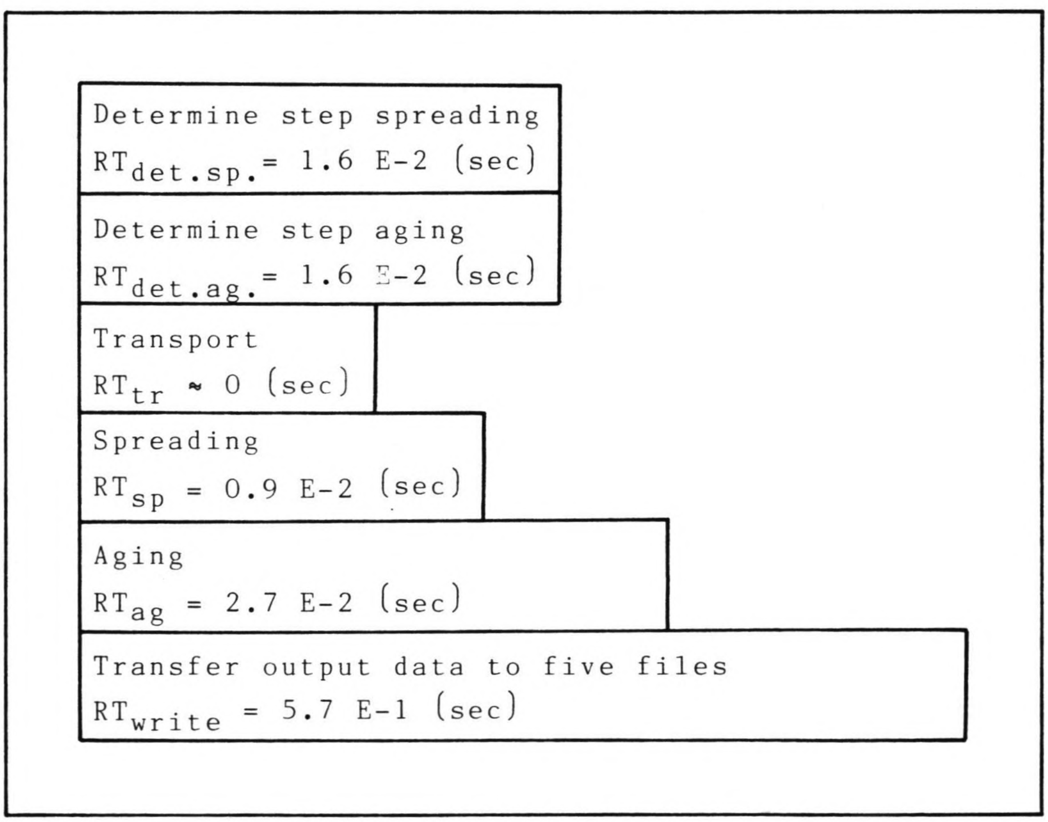


Figure 7.47. Computer run time six elements process block.

Example calculation of simulation run time:

A twelve hour simulation period and sixty seconds integration step-sizes are assumed.

[1] RT_{init}	$= 11.7 + 3.6 * 1$	$= 15.3$
[2] $RT_{det.sp.}$	$= 719 * 1.6\ E{-2}$	$= 11.5$
$RT_{det.ag.}$	$= 719 * 1.6\ E{-2}$	$= 11.5$
[3] RT_{tr}	$= 720 * 0$	$= 0.0$
[4] RT_{sp}	$= (60 + 12 + 3 + 717 * 1) * 0.9\ E{-2}$	$= 7.1$
RT_{ag}	$= (60 + 12 + 3 + 717 * 1) * 2.7\ E{-2}$	$= 21.4$
[5] RT_{write}	$= (12 * 2) * 5.7\ E{-1}$	$= 13.7$
RT_{term}	$= 0$	$= 0.0$ +
		80.5 (seconds)

[1]The twelve hour period is assumed to be placed within one day (e.g. 00.00 - 12.00) and therefore the number of interpolated days is one.

[2]The determination of the step-size takes place each minute with exception of the first. A twelve hour period equals 720 minutes minus 1 is 719.

[3]The transport is calculated each minute.

[4]The initial integration step-size is one second and therefor 60 calculations the first minute (also see figure 5.6.). The second minute (as is assumed) twelve calculations with a step-size of five seconds. The third minute three calculations and from the fourth minute just one calculation for each minute.

[5]A thirty minutes output interval is assumed.

Under the above mentioned conditions the run speed of the process block is: (80.5 - 15.3) / 12 = 5.4 (seconds/hour)

7.6. Efficiency

Efficiency: - ability to produce the desired effect with a minimum of effort
- the ratio of effective work to the energy expended in producing it

Efficiency can - in the MS4 context - be interpreted as the tradeoff between run time and accuracy. Aim of effeciency is a minimal computer run time but with a still acceptable accuracy loss.

Accuracy loss: The loss of accuracy - going from SMOSS to MS4 - is only caused by a larger integration step resulting in a larger truncation error.

A numerical indication of this loss of accuracy is achieved by dividing the MS4 results by the SMOSS results.

Acceptable : As has been written in paragraph 7.2. the accuracy is the sum of several types of errors. In this case the loss of accuracy is caused by a larger truncation error. So to judge whether this loss of accuracy is acceptable it should be compared to the original accuracy (i.e. the original sum of errors).

Efficiency is determined by computer run time and accuracy. Therefore the influence of several variables on run time and accuracy should be investigated.

Determining for run time T_{run}:

- $\Delta t_{sp(reading)}$: integration step size spreading process see figure 7.48.

- $\Delta t_{ag(ing)}$: integration step size aging process see figure 7.48.

- $\Delta t_{out(put)}$: time step used to write results to files see figure 7.49.

- $T_{sim(ulation)}$: simulated time period see figure 7.52

- available storage space; see figures 7.50. and 7.51.

- use of background processing (spawning); see figure 7.53.

* the integration step of the transport process is fixed at one minute and can therefore not influence run time; the reason to choose an one minute time step has been explained in paragraph 5.5.

Determining for accuracy:

- $\Delta t_{sp(reading)}$: integration step size spreading process see figure 7.55.

- $\Delta t_{ag(ing)}$: integration step size aging process see figures 7.56., 7.57. and 7.58.

- data type : REAL*4 or REAL*8 see figure 7.54.

* the integration step of the transport process is fixed at one minute and can therefore not influence the accuracy of the transport variables; the reason to choose an one minute time step has been explained in paragraph 5.5.

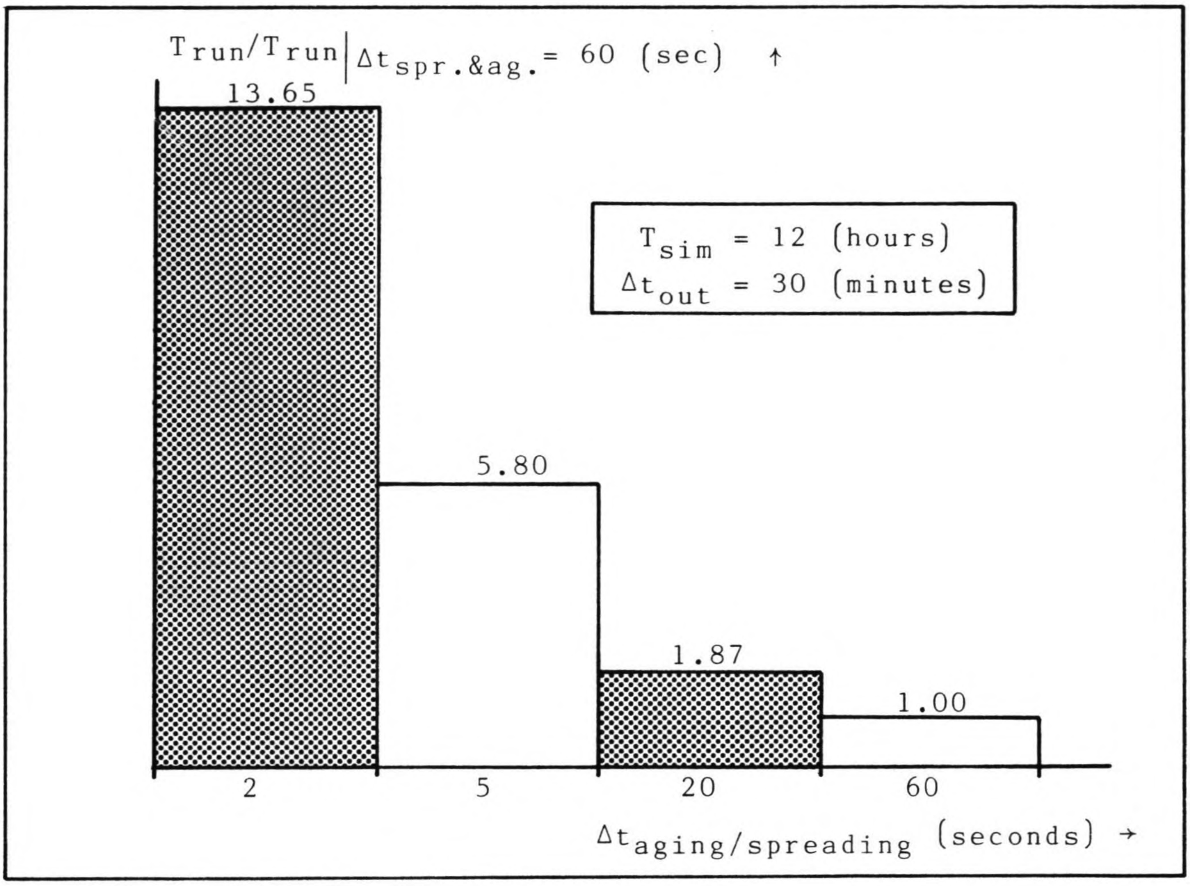


Figure 7.48. T_{run} as function of $\Delta t_{aging/spreading}$.

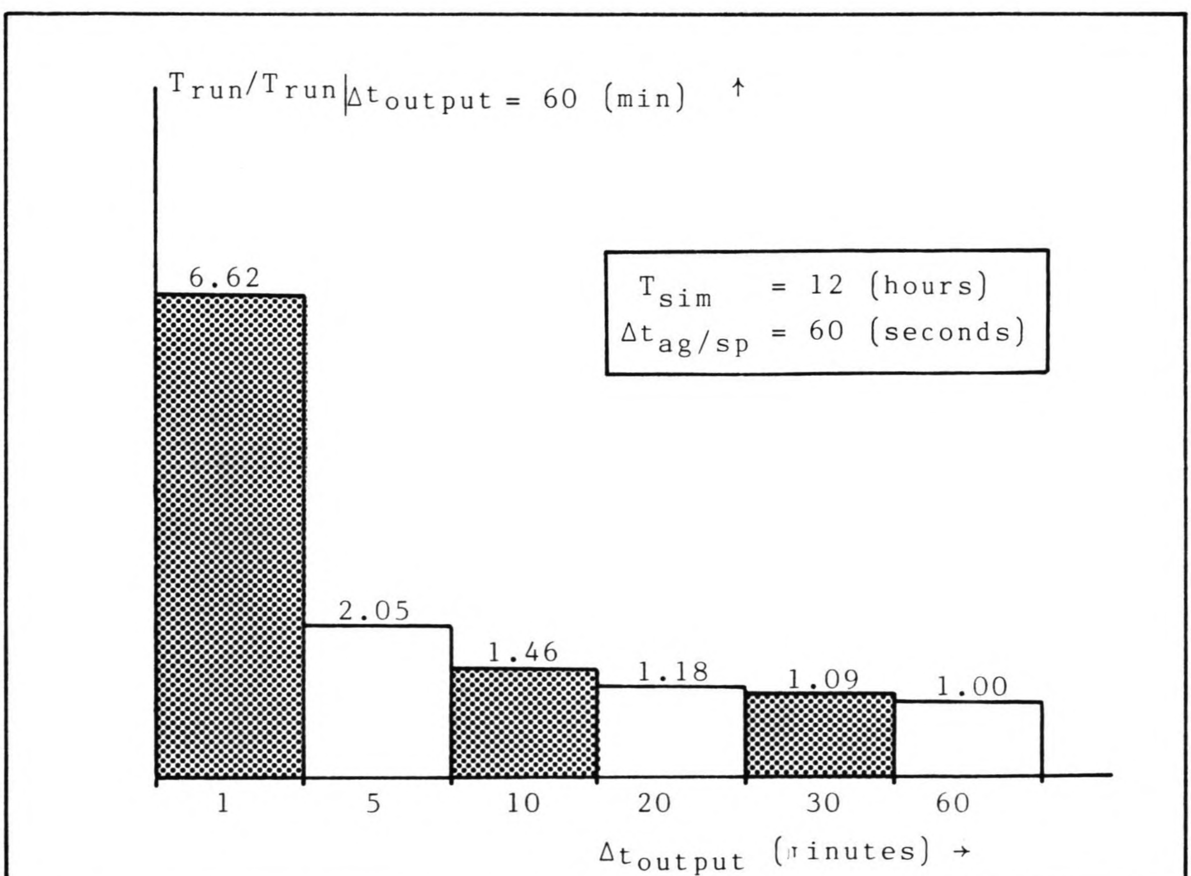


Figure 7.49. T_{run} as function of Δt_{output}.

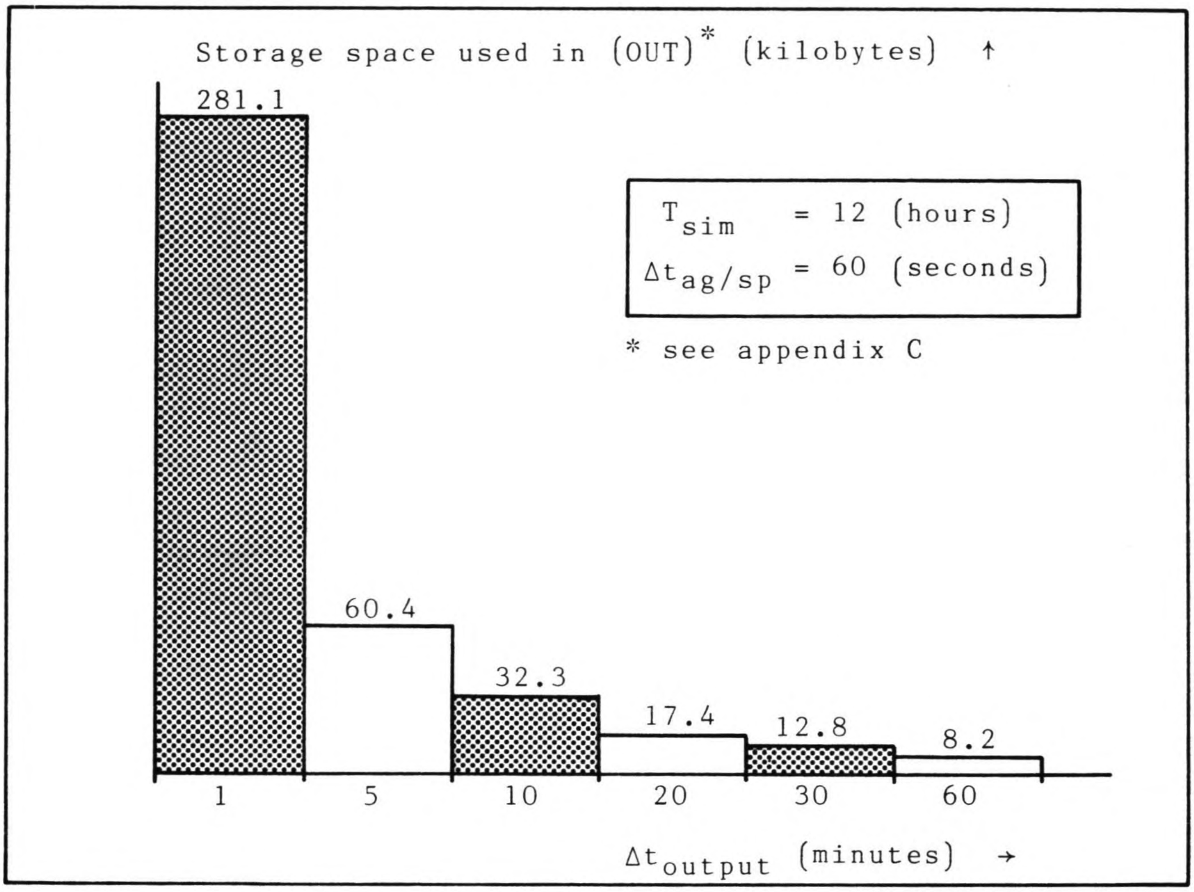


Figure 7.50. Used storage space as function of Δt_{output}.

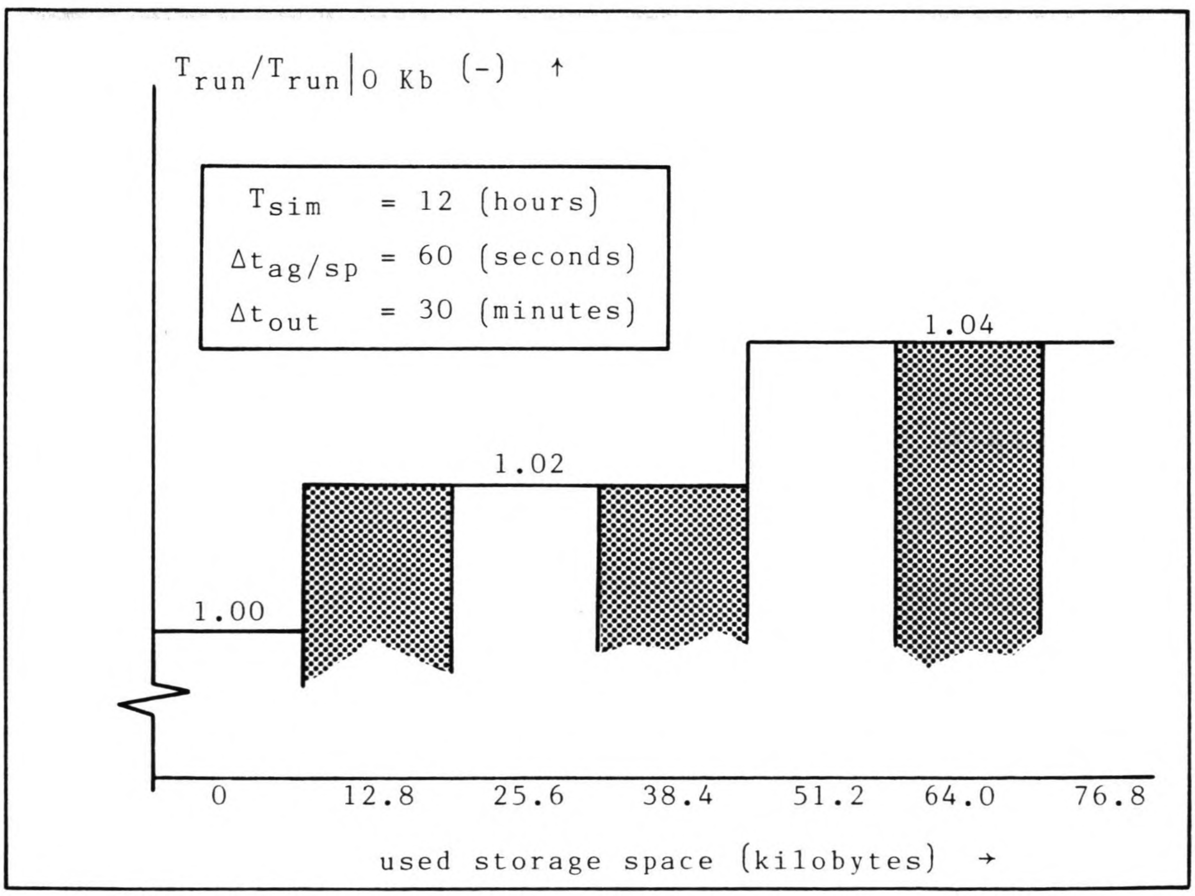


Figure 7.51. T_{run} as function of the used storage space in (OUT).

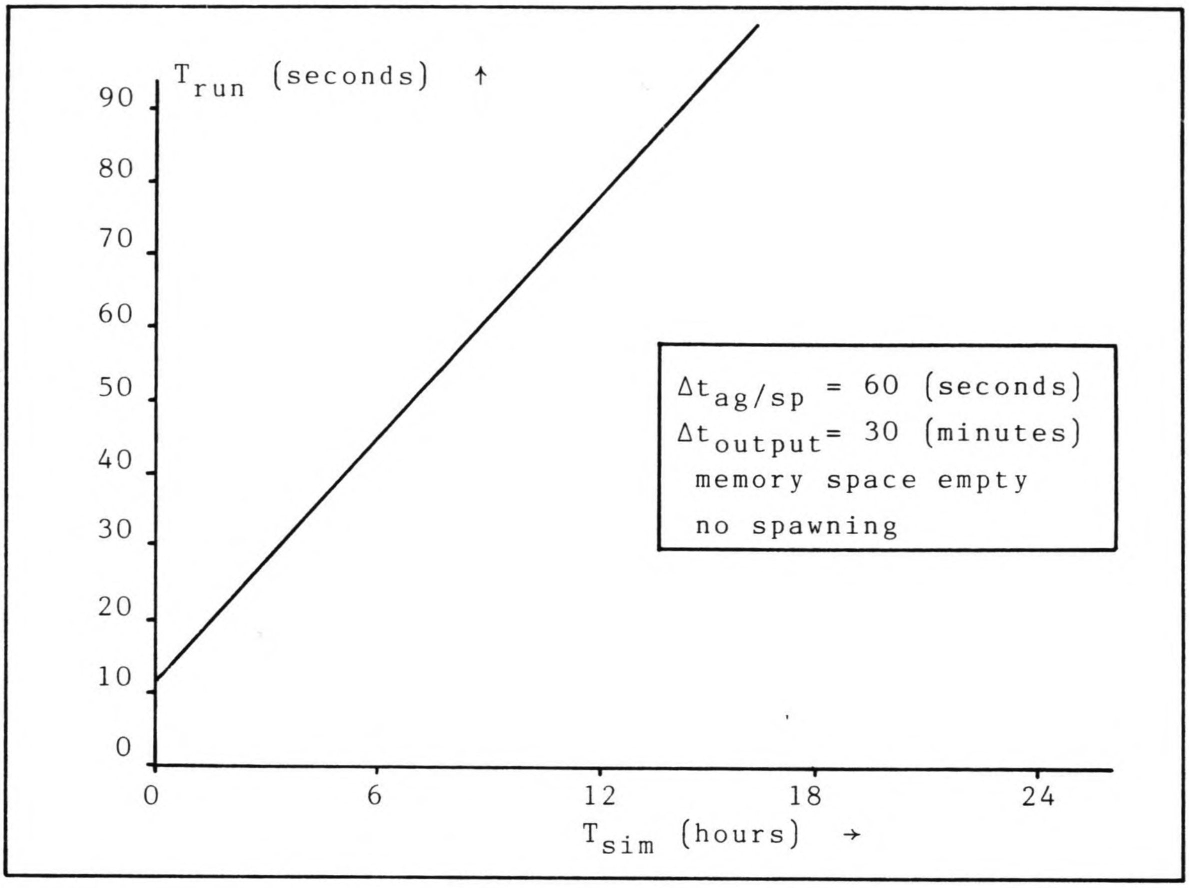


Figure 7.52. T_{run} as function of T_{sim}.

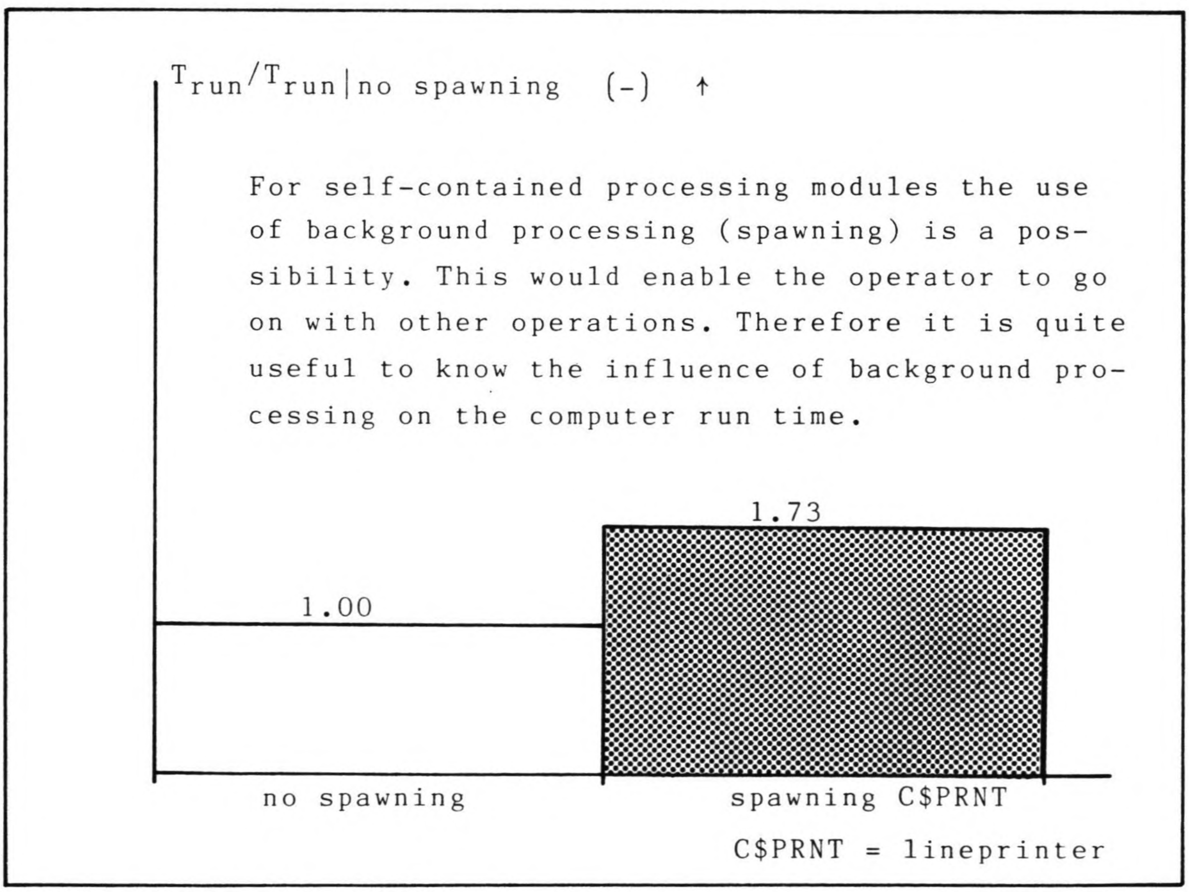


Figure 7.53. T_{run} as function of spawning.

Instead of using REAL*4 numbers, the use of REAL*8 numbers has been studied. The influence on the task image size can be seen in figure 7.54..

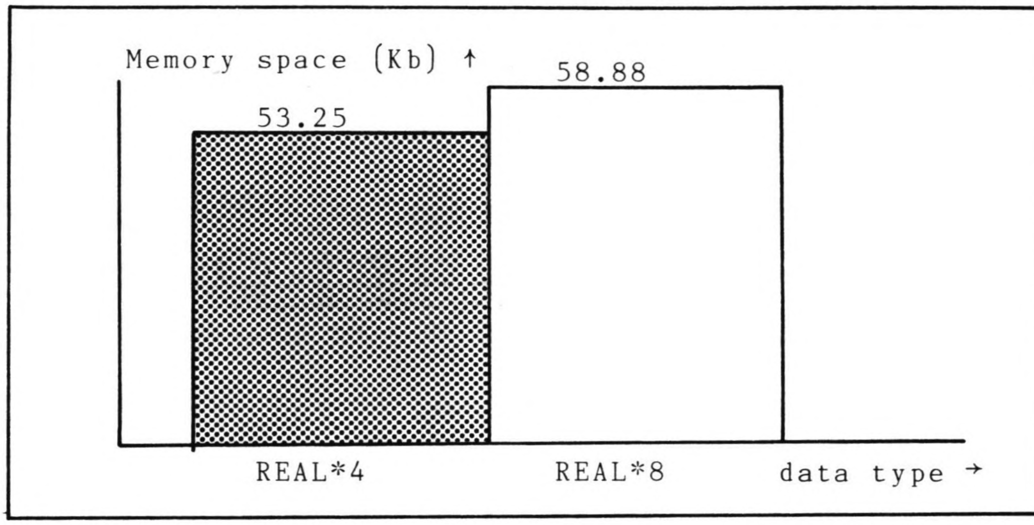


Figure 7.54. REAL*4 versus REAL*8

It was not possible to run the double precision version because not enough free memory was left in the OTS buffer area to set up required I/O control blocks and buffers.

However, knowing that the model error is considerable, it is absolutely superfluous to compute the simulation with double precision numbers.

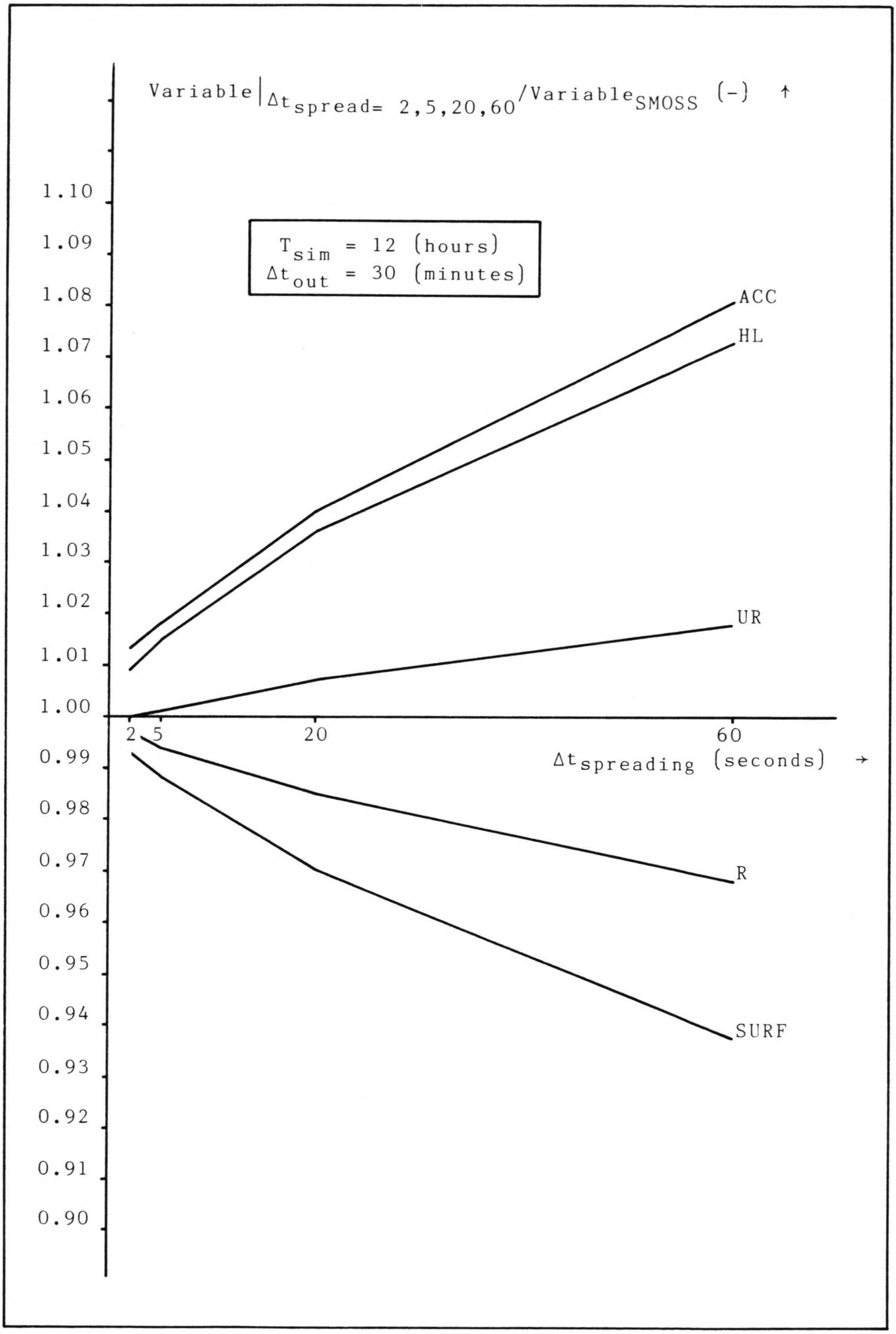


Figure 7.55. Accuracy spreading variables as function of the integration step $\Delta t_{spreading}$.

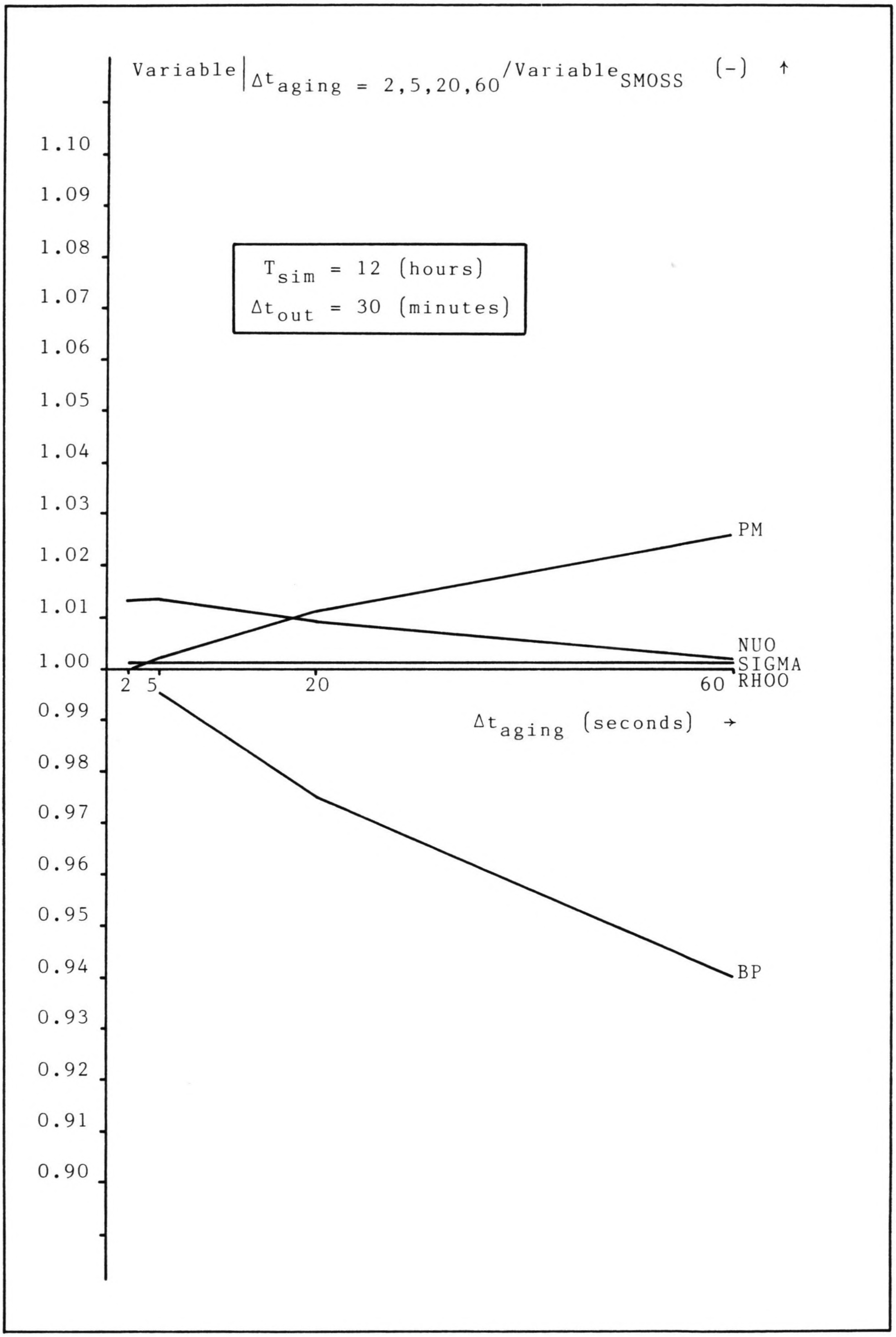


Figure 7.56. Accuracy aging variables as function of the aging integration step Δt_{aging}.

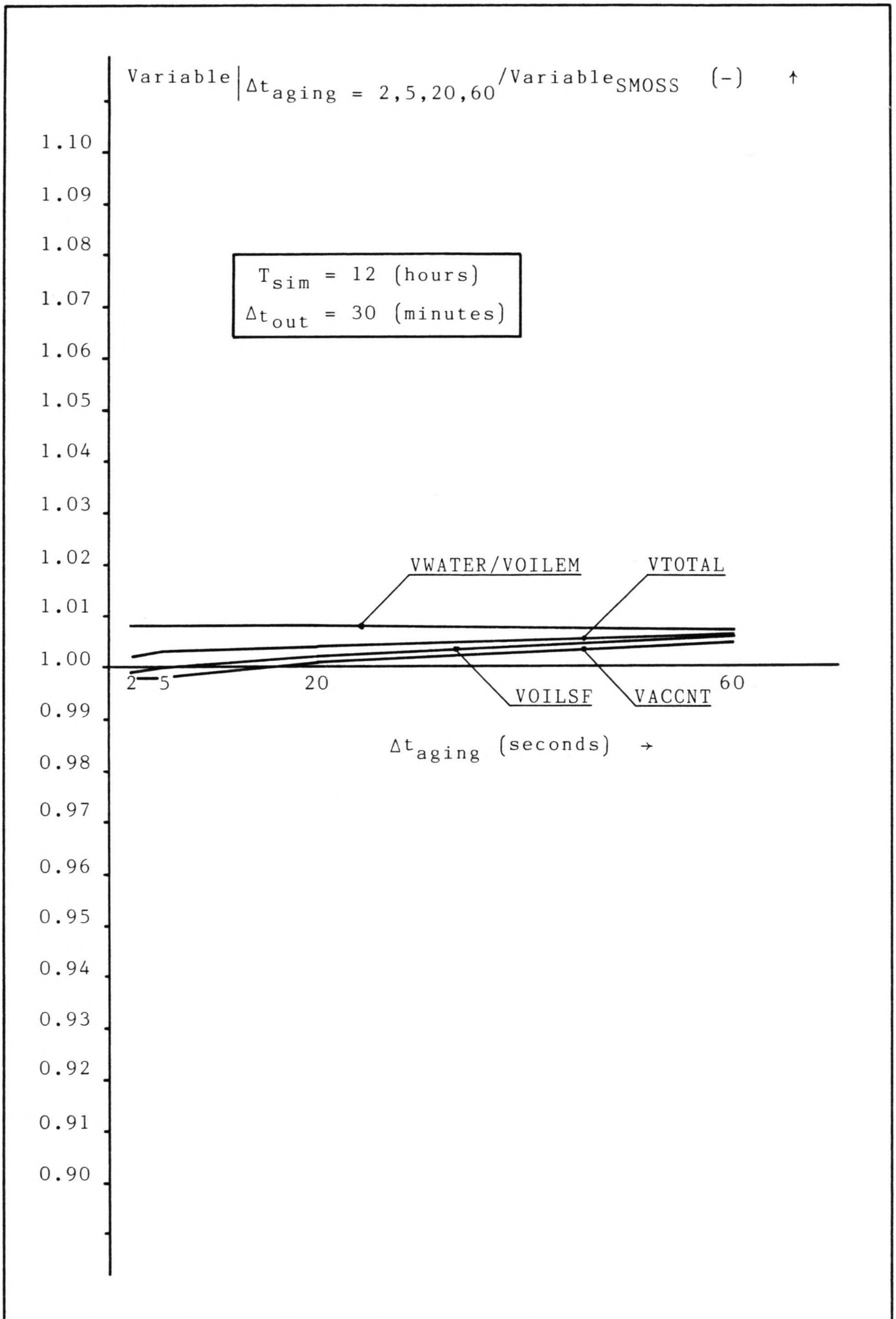


Figure 7.57. Accuracy aging variables as function of the aging integration step Δt_{aging}.

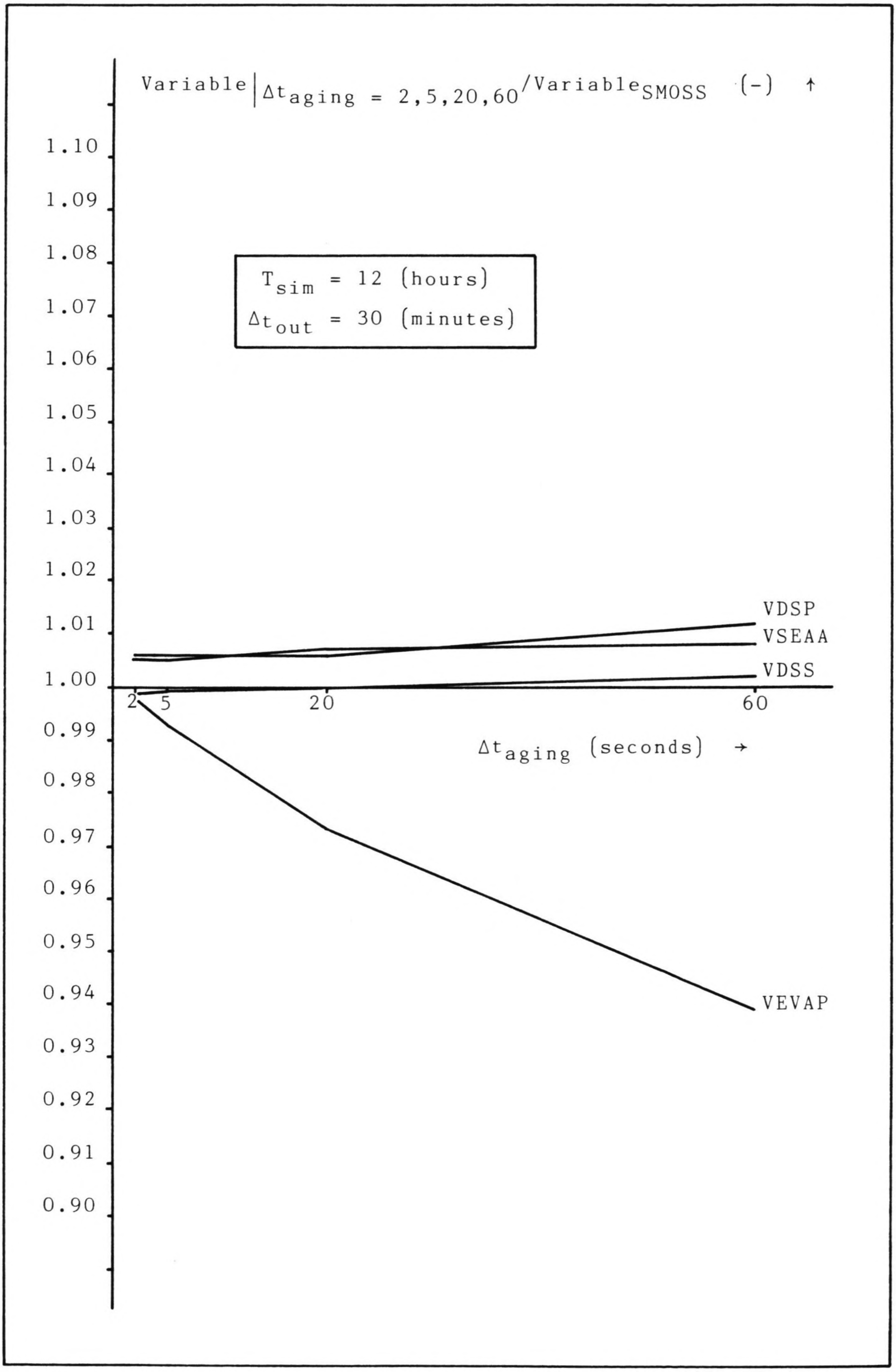


Figure 7.58. Accuracy aging variables as function of the aging integration step Δt_{aging}.

7.7. Stability

Stability concepts are suggested by physics, where stabilty means, roughly speaking, that a small change - small disturbance - of a physical system at some instant changes the behavior of the system only slightly at all future times.

To check MS4 stability several disturbances of the simulation input are imaginable. Singled out for a check on stability is an abrupt change of the wind-velocity.

In order to facilitate the assesment of results all other input variables equal those as used in the program testing; see paragraph 7.2. and 7.3..

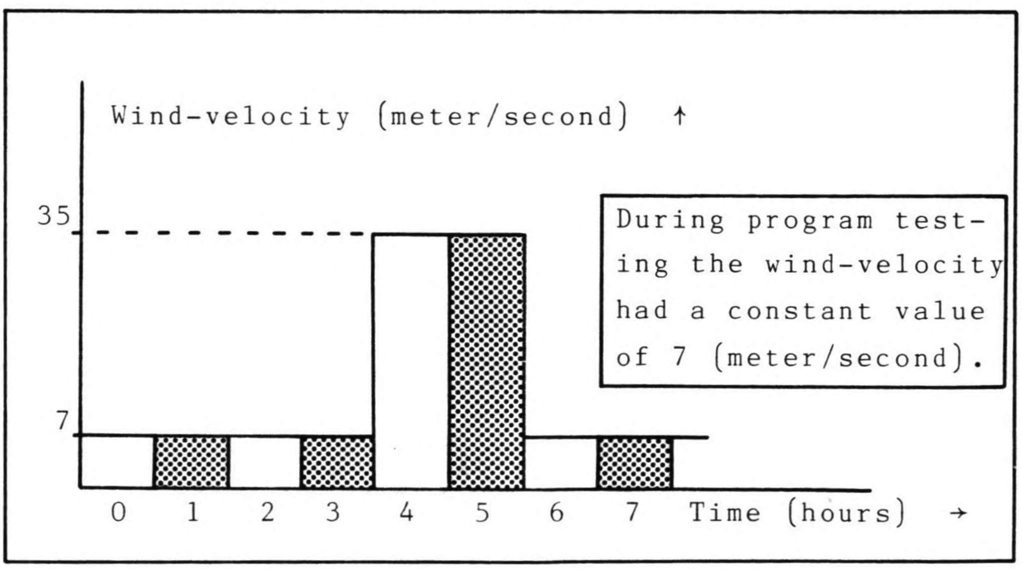


Figure 7.59. An abrupt change of the wind-velocity.

Because the disturbance is only present at the fourth and fifth hour of the simulation a twenty hour simulation output period will suffice to judge whether the system is stable or not.

To judge the stability of the model the two following runs are compared:

1 The test run (see appendix J).
2 The "stability" run, which equals the test run with the important exception that the wind-speed has an abrupt two hour change.

The behavior of the physical system can be classed in three categories: transport, spreading and aging.

Transport:

A change in wind-speed directly influences the transport. This will result in a diverging transport curve during this two hour period. Outside this time period the slick velocity of the stability run will exactly equal the transport of the test run. The transport system is stable. What's more: instability is impossible! The reason for this guaranteed stablity lies in the fact that the transport is completely independent of the aging and spreading process. There is no interaction possible between two processes.

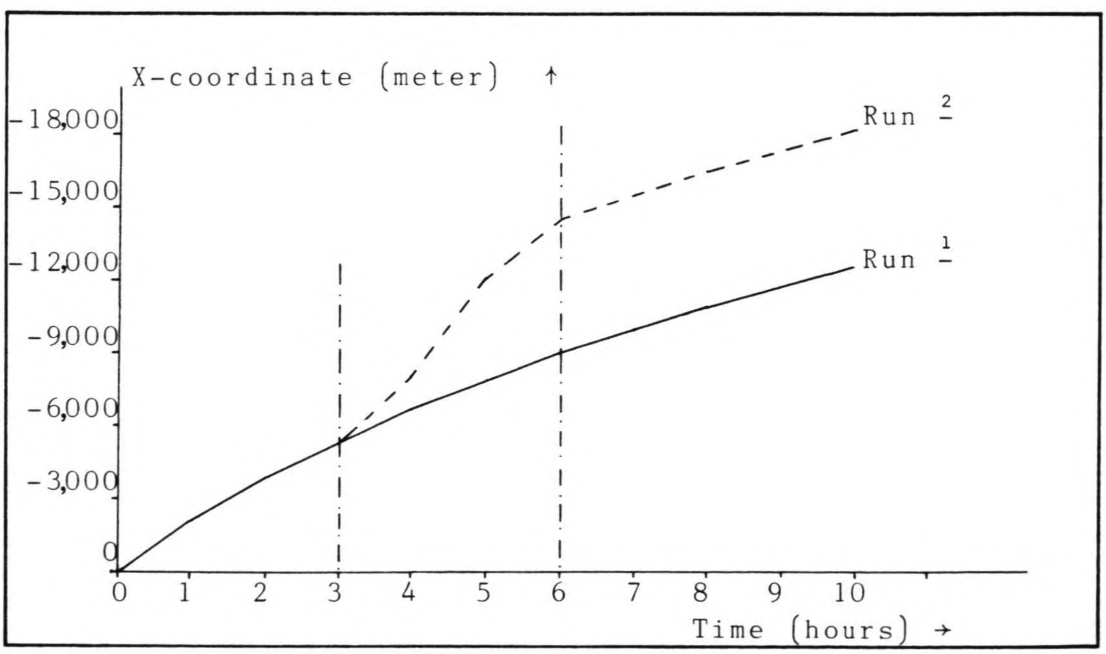


Figure 7.60. Stability transport process; comparison between test run and stability run.

Spreading/aging:

The wind-speed influences directly the rate of evaporation DVEVAP - an element of the aging process - and therefore the evaporated volume VEVAP. Thus - by the balance of volumes - the whole aging process is influenced by a change of the wind-speed. The coupling between the aging and spreading process causes a change of the spreading variables. So a certain interaction is present!

Now the question rises whether this abrupt change of the wind-speed will cause - through the existing interaction - an amplification effect. In other words: will it lead to instability?

To avoid unnecessary paperwork the stability test result will suffice: no instabilities have been detected!

This stability test is the only one so far although several other stability tests are imaginable. However, the reaction of the system to an abrupt change of wave-height, wave-period, tidal-speed or current-speed will be virtually analogous or smaller than that caused by the wind-speed change.

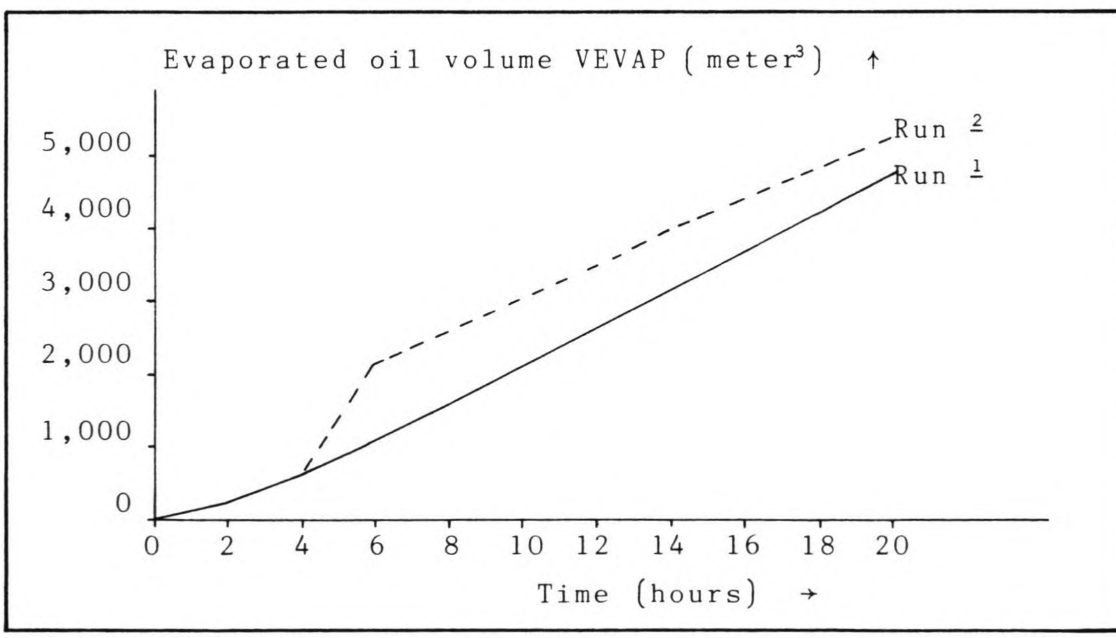


Figure 7.61. Stability aging process; comparison between test run and stability run.

8. Conclusions and recommendations

8.1. Conclusions

The programming language of the simulation model has been converted from CSMP III to FORTRAN 77.

The loss in run speed by switching from a mainframe to a microcomputer has almost entirely been compensated by more economical routines, a modular structure and a different programming language.

The user friendly interaction between operator and computer is a salient feature of the simulation.

The output is neatly arranged and can easily be obtained at any phase of the simulation.

The discrepancy in results between SMOSS and MS4 is in the range 0% - 10% with a single relatively unimportant exception.

The simulation model shows no signs of instability.

8.2. Recommendations

In the course of this thesis especially the computer technological aspects of the model have been studied and improved. So the foundations of the simulation structure have been well laid for the coming years.

Therefore it is recommended - in case of a continuance of this project, which is to be expected - to bring one's mind to bear upon the problem of improving the physical description and translating it into a mathematical description.

Furthermore it is recommended to include combat strategies in the simulation model.

Recommendations, related to this work, are:

- Change the data accumulation source programs such that the data files in directory (EXTRA) can be deleted.

- Add help texts for each menu and for each option.

- Add single line warning for each input field of the data accumulation form.

- Add a source program that provides a visual screen display of the map and the slick.

- Add data bases for regular returning data.

- Consider to leave the integration step-size determination out. This would speed up run time by a factor 2. This is probably possible without leading to too large inaccuracies.

- Add visual display of output data.

- Consider the use of background processing.

Notation

Symbol	Definition
ACC	acceleration of spreading
AFGEN	CSMP interpolation method
BP	boiling point
d	distance
DVEVAP	dVEVAP/dt
f	integral curve
h	integration step size
H	wave-height
HL	oil layer thickness
n	order polynomial equation
NUO	oil viscosity
PM	vapor pressure * molecular weight
R	slick radius
RT	run time
SC	slick-center
SIGMA	net surface tension
STIFF	CSMP integration method
SURF	oil surface area
t	time
T_{sim}	simulated time period
T_{run}	computer run time simulation
UR	slick spreading velocity
VACCNT	volume oil accent
VDSP	dispersed oil volume
VDSS	dissolved oil volume
VEVAP	evaporated oil volume
VOILEM	emulsified oil volume
VOILSF	volume oil at the surface
VSEAA	volume oil sea-air exchanged
VTOTAL	volume oil plus volume water
VWATER	volume water in oil
X	X co-ordinate slick-center
Y	Y co-ordinate slick-center

Symbol	Definition
Δt_{transp}	integration step-size transport process
$\Delta t_{sp(reading)}$	integration step-size spreading process
$\Delta t_{ag(ing)}$	integration step-size aging process
$\Delta t_{out(put)}$	time step that output is written to files
ξ	point in integration interval
$\int$	symbol of integration
$'$	symbol of derivative

References

1. Kuipers, H. D.
   Processes which influence the motions
   and characteristics of oil at sea.
   Delft University of Technology and
   North Sea Directorate
   Ministry of Transport and Public Works, Rijswijk,
   The Netherlands. January 1980.

2. Kuipers, H. D.
   A simulation model for oil slicks at sea.
   Delft University of Technology, Delft
   North Sea Directorate
   Ministry of Transport and Public Works, Rijswijk,
   The Netherlands. January 1980.

3. Kammer, Ir. R.
   Numerieke methoden voor technici
   Elsevier Amsterdam/Brussel 1977

4. Digital Equipment Corporation
   Reference Manuals Professional 350

5. Curran, Susan and Curnow, Ray
   The penguin computing book
   A complete and comprehensive guide to computing
   Great Britain 1983

Appendix A: A survey of the simulation processes

This thesis deals primarily with information theoretical aspects of the simulation model. However, physical and information theoretical aspects are always closely linked. Therefore a closer look at the physical processes, underlying the simulation, is well justified. This appendix is based on the engineers thesis of Herman D. Kuipers (see References).

Potentialities of the simulation:

The following processes can be simulated:
- Transport
- Spread
- Aging

Transport:

Slick transport is caused by:
- Waves
- Winds
- Tides
- Currents

Waves : $u_{slick} = u_{waves} = \dfrac{(\pi H)^2}{(1.56 * T^3)}$; mass transport velocity

Winds : $u_{slick} = 0.0283 * u_{wind}$

Tides : $u_{slick} = u_{tide}$

Currents: $u_{slick} = u_{current}$ +

$$u_{slick} = u_{waves} + 0.0283 * u_{wind} + u_{tide} + u_{current}$$

$$r_{slick}\big|_{t+\Delta t} = r_{slick}\big|_{t} + \int u_{slick} \cdot dt$$

Notation: See final pages of this appendix.

Spread:

A cylindrical oil slick is assumed.
Spread due to five forces; see figure A.1..

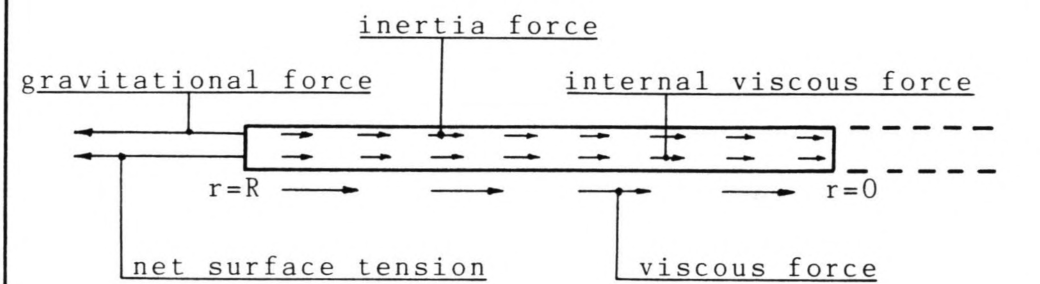


Figure A.1. Five forces spreading process.

1. Gravitational force : $F_{gra} = \Delta V^2 \rho_0 g / \pi R^3$

2. Net surface tension : $F_{nst} = 2\pi R\sigma$

3. Inertia force : $F_{ine} = \frac{2}{3}\pi h \rho_0 \left(\frac{d^2R}{dt^2}.R^2 - \left(\frac{dR}{dt}\right)^2.R\right)$

4. Viscous force : $F_{vis} = 1.45\rho_w \nu_w^{\frac{1}{2}} R^{1\frac{1}{2}}.\left(\frac{dR}{dt}\right)^{1\frac{1}{2}}$

5. Internal viscous force: $F_{ivs} = \pi \nu_0 h R \frac{dR}{dt}$

Balance of forces :

$$\frac{d^2R}{dt^2} = \frac{3}{2}.\frac{\Delta V g}{\pi R^3} + \frac{3\pi R\sigma}{\rho_0 V} + \frac{1}{R}\left(\frac{dR}{dt}\right)^2 - \frac{2.175\rho_w \nu_w^{\frac{1}{2}} R^{1\frac{1}{2}}}{\rho_0 V}\left(\frac{dR}{dt}\right)^{1\frac{1}{2}} - \frac{3}{2}.\frac{\nu_0}{R^2}.\frac{dR}{dt}$$

Notation: See final pages of this appendix.

Aging:

The simulation includes the following aging processes:

- Evaporation
- Emulsification
- Dispersion
- Dissolution
- Direct sea-air exchange

Evaporation : $$\frac{dVevap}{dt} = k_4 * PM * R^{2-\beta}$$

$$k_4 = k_{18} * FRACTN * 0.031 * Kev * 2^{2-\beta} * u_{wind}^{\alpha}$$

$$PM\Big|_{t+t} = PM\Big|_t + \frac{PM_{init} * \Delta V_{evap}}{FRACTN * Vaccnt}$$

Emulsification: $$Voilem = Voilsf - Vaccnt$$

$$Vaccnt = Voilin + \int_{t_0}^{t} \frac{dVaccnt}{dt} . dt$$

$$Voilsf = Voilin - (Vevap + Vsea\text{-}air + Vdisp + Vdiss$$

$$\frac{dVaccnt}{dt} = - \frac{Voilsf}{t_{end}} + \frac{dVoilsf}{dt}$$

$$\frac{dVoilsf}{dt} = - \left(\frac{dVevap}{dt} + \frac{dVsea\text{-}air}{dt} + \frac{dVdisp}{dt} + \frac{dVdiss}{dt} \right)$$

$$t_{end} = k_{14} / (EMSCST * H)$$

Dispersion : $$\frac{dVdisp}{dt} = k_{15} * k_{16} * Vaccnt * H$$

$$Vdisp = \int_{t_0}^{t} \frac{dVdisp}{dt} . dt$$

Notation: See final pages of this appendix.

Dissolution : $$\frac{dVdiss}{dt} = (c_6 + c_3) * Vaccnt$$

$$Vdiss = \int_{t_0}^{t} \frac{dVdiss}{dt}.dt$$

Direct sea-air exchange: $$\frac{dVseaair}{dt} = c_{15} * Vaccnt * H$$

$$Vseaair = \int_{t_0}^{t} \frac{dVseaair}{dt}.dt$$

The simulation includes changes in:

- Flashpoint
- Oil density
- Oil viscosity
- Net surface tension

Flashpoint : $FLASHP = 0.73 * BP - 72.6$ in ^0C

$$BP\Big|_{t+\Delta t} = BP\Big|_{t} + k_{17} * \frac{\Delta Vevap}{Vaccnt}$$

Oil density : $$\frac{d\rho_0}{dt} = k_{88} + k_{89} * \frac{dVevap}{dt} / Voilsf$$

$$+ \left[\frac{dVwater}{dt} * (\rho_w - \rho_0(t)) + \frac{dVoilsf}{dt} * (\rho_0(t_0) - \rho_0(t))\right] / Vtotal(t)$$

$$\rho_0 = \rho_0(t_0) + \int_{t_0}^{t} \frac{d\rho_0}{dt}.dt$$

$$Vwater = 4 * Voilem$$

Oil viscosity : $\nu_0 = \nu_{init} * (1 - Vwater / Vtotal)^{-4.0}$

$$Vtotal = Voilsf + Vwater$$

Net surface tension: $\sigma = \sigma_{init} * (1 - Voilem / Voilsf)$

The simulation includes the time at which:

- the oil will sink
- the oil spill will break up

Sinking of oil : Oil sinks if $\rho_{oil} > \rho_{water}$.

Break up of oilslick: $t_{break\ up} = (Voilsf)^{0.33} * 10^{+4}$ (seconds)

Notation exclusively for appendix A:

Symbol	Definition
c	constants
F	force
F_{gra}	gravitational force
F_{nst}	net surface tension
F_{ine}	inertia force
F_{vis}	viscous force
F_{ivs}	internal viscous force
g	gravitational acceleration
h	oil layer thickness
H	wave height
K_{ev}	evaporation constant
M	molecular weight
n	turbulence parameter
P	vapor pressure
r	radius
R	slick radius
r_{slick}	slick position
t	time
u	velocity
u_{slick}	slick velocity
u_{waves}	slick velocity caused by waves
u_{winds}	wind velocity
u_{tides}	tidal velocity
$u_{currents}$	current velocity
V	volume
α	$(2 - n) * (2 + n)$
β	$n / (2 + n)$
Δ	$(\rho_w - \rho_0) / \rho_w$
μ	dynamic viscosity
ν	kinematic viscosity

Symbol	Definition
π	3.14159
ρ	density
$\Delta\rho$	$\rho_w - \rho_0$
σ	surface tension
$_0$	oil
$_w$	water

The remaining undefined symbols can either be found in the general notation or in the listing of the source program.

Appendix B: Interaction user - computer

B.1. Introduction

The Marine Spill Simulation Software Set consists of the three following installed applications:

a. MS4-1: Data accumulation and processing
b. MS4-2: Actual simulation
c. MS4-3: Output presentation

This sequence (MS4-1 → MS4-2 → MS4-3) must be maintained in executing a new simulation.

In describing the interaction between user and computer the following expression and signs have been used:

Level
The expression "level" has been used to distinguish seperate parts of the application. In practice a new level coincides with a new menu.

-
Indicates one option of the displayed menu.

.....
Indicates one or more options not relevant to MS4.

!
Indicates an option which the user must choose prior to all other options.

←
After execution of the task this option stands for, control will be transferred to the preceding menu.

*
After execution of the task this option stands for, control will be transferred to the same menu.

→

After execution of the task this option stands for, control will be transferred to the next menu.

∇

Indicates a dummy option added to the menu in view of possible later additions.

e

Indicates an "Exit" option which can be used to leave the MS4 application.

B.2. MS4-1: Data accumulation and processing

Users action	Computers action
Switch on computer and lineprinter	
	Display main menu P/OS V1.5 Eight options: - Prose - Basic - ..... - Extra applications - .....
! Select "Extra applications"$^{\lambda}$	
	Display menu with the application groups. Eleven options: - A Group - B Group - C Group - ....
! Select "B Group"	

-------------------------Level 1-------------------------------

Users action	Computers action
	Display menu with the MS4 applications. Three options: - MS4-1: Data accumulation and processing - MS4-2: Actual simulation - MS4-3: Output presentation
! Select "MS4-1: Data accumulation and processing"	

ad λ In the present-day simulation the directory MS4 has to be selected prior to this step. This condition - to select the right directory - will be eliminated in the near future.

Users action	Computers action
------------------------Level 2	------------------------------
	Display MS4-1 menu "MAIN". Three options: - Prepare input for new simulation - Modify input previous simulation - Exit
→ Select "Prepare input for new simulation"	
	Go to level 3.
∇ Select "Modify input previous simulation"	
	Return to level 2.
e Select "Exit"	
	Exit MS4-1. Return to level 1.
------------------------Level 3	------------------------------
	Display MS4-1 menu "NEW". Twelve options: - Time - Geography - Slick - Oil properties - Sea water - Weather - Winds - Waves - Tides - Currents - Combat strategies - Proceed

Users action	Computers action
* Select "Time"	
	Call subroutine TIMES. Gathers all data concerning time interval of interpolation and simulation. All data are written to the following files: DW1:(DF)INTER.DAT DW1:(DF)SIMUL.DAT Return to level 3.
* Select "Geography"	
	Go to level 3.1..

------------------------Level 3.1.----------------------------

	Display MS4-1 menu "GEOGR" Three options: - Southern part Northsea - Previous seachart - New seachart
∇ Select "Southern part Northsea"	
	Return to level 3.1.. In the future the file belonging to the option chosen will be written to: DW1:(DF)GEOGR.DAT
∇ Select "Previous seachart"	
	Return to level 3.1.. In the future the file DW1:(DF)GEOGR.DAT will be maintained.
! Select "New seachart"	
	Calls subroutine GEOGR. Gathers all data necessary to plot a seachart. All data are written to file: DW1:(DF)GEOGR.DAT Return to level 3.

Users action	Computers action
* Select "Slick"	
	Call subroutine SLICK. Gathers all data concerning slick properties. All data are written to the file: DW1:(DF)SLICK.DAT Return to level 3.
* Select "Oil properties"	
	Go to level 3.2..

------------------------Level 3.2.-----------------------------

	Display MS4-1 menu OIL. Five options: - Light Arabian Crude - Residual Oil - Gasoline - Light Naphta - Other oil type
∇ Select "Light Arabian Crude" or "Residual Oil" or " Gasoline" or " Light Naphta"	
	Return to level 3.2.. In the future the file belonging to the option chosen will be written to: DW1:(DF)OILPR.DAT
! Select "Other oil type"	
	Call subroutine OILPR. Gathers all data about oil properties. All data are written to the file: DW1:(DF)OILPR.DAT Return to level 3.

Users action	Computers action
* Select "Sea water"	Call subroutine SEAWA. Gathers seawater data and are written to file: DW1:(DF)SEAWA.DAT Return to level 3.
* Select "Weather"	Call subroutine WEATH. Gathers weather data and these are written to file: DW1:(DF)WEATH.DAT Return to level 3.
* Select "Winds"	Call subroutine WINDS. Gathers wind data at arbitrary points of time and are written to the file: DW1:(SDF)WINDS.DAT Return to level 3.
* Select "Waves"	Call subroutine WAVES. Gathers wave data at arbitrary points of time and are written to the file: DW1:(SDF)WAVES.DAT Return to level 3.
* Select "Tides"	Call subroutine TIDES. Gathers tidal data at arbitrary points of time and are written to the file: DW1:(SDF)TIDES.DAT Return to level 3.

Users action	Computers action
* Select "Currents"	
	Call subroutine CURRE. Gathers current data at arbitrary points of time and are written to the file: DW1:(SDF)CURRE.DAT Return to level 3.
∇ Select "Combat strategies"	
	Call subroutine COMBA. Show a form with information about future combat possi-bilities.
→ Select "Proceed"	
	Go to level 4.

------------------------Level 4--------------------------------

	Display MS4-1 menu LIST1. Four options: - Listing of all input files - Proceed with interpolation - By-pass the interpolation - Exit
* Select "Listing of all input files"	
	Call subroutine LIST1. Print the following files: DW1:(DF)INTER.DAT DW1:(DF)SIMUL.DAT DW1:(DF)SLICK.DAT DW1:(DF)OILPR.DAT DW1:(DF)SEAWA.DAT DW1:(DF)WEATH.DAT DW1:(DF)GEOGR.DAT DW1:(SDF)WINDS.DAT DW1:(SDF)WAVES.DAT DW1:(SDF)TIDES.DAT DW1:(SDF)CURRE.DAT Return to level 4.

Users action	Computers action
! Select "Proceed with interpolation"	Call subroutines: INTWND INTWAV INTTID INTCUR These subroutines interpolate data from the following files: DW1:(SDF)WINDS.DAT DW1:(SDF)WAVES.DAT DW1:(SDF)TIDES.DAT DW1:(SDF)CURRE.DAT By interpolating the data from these files new data files can be written with time intervals of one hour. These files are: DW1:(DF)WINDS.DAT DW1:(DF)WAVES.DAT DW1:(DF)TIDES.DAT DW1:(DF)CURRE.DAT Then go to level 5.
→ Select "By-pass the interpolation"	Go to level 5. This is only possible if interpolated data files are already available.
e Select "Exit"	Return to level 1.

Users action	Computers action
--------------------------Level 5--------------------------	
	Display MS4-1 menu LIST2. Three options: - Listing of all interpolated input files - Return to the main menu - Exit
* Select "Listing of all interpolated input files"	
	Call subroutine LIST2. Print the following files: DW1:(DF)WINDS.DAT DW1:(DF)WAVES.DAT DW1:(DF)TIDES.DAT DW1:(DF)CURRE.DAT Return to level 5.
* Select "Return to the main menu"	
	Go to level 2.
! Select "Exit"	
	Go to level 1.

B.3. MS4-2: Actual simulation

Users action	Computers action
Switch on computer and lineprinter	
	Display main menu P/OS V1.5 Eight options: - Prose - Basic - ..... - Extra applications - .....
! Select "Extra applications" $^{\lambda}$	
	Display menu with the application groups. Eleven options: - A Group - B Group - C Group - ....
! Select "B Group"	

-------------------------Level 1------------------------------

Users action	Computers action
	Display menu with the MS4 applications. Three options: - MS4-1: Data accumulation and processing - MS4-2: Actual simulation - MS4-3: Output presentation
! Select "MS4-2: Data accumulation and processing"	

ad λ In the present-day simulation the directory MS5 has to be selected prior to this step. This condition - to select the right directory - will be eliminated in the near future.

Users action	Computers action
--------------------------Level 2	-------------------------------
	Display MS4-2 menu "SEC1". Two options: - Read all required input variables - Exit
→ Select "Read all required input variables"	
	Read all data from the following data files: DW1:(DF)SIMUL.DAT DW1:(DF)INTER.DAT DW1:(DF)SLICK.DAT DW1:(DF)OILPR.DAT DW1:(DF)SEAWA.DAT DW1:(DF)WEATH.DAT Read only number of interpolated days from file: DW1:(DF)WAVES.DAT Give parameters their value and give variables their initial value. Proceed to level 3.
e Select "Exit"	
	Exit MS4-2. Return to level 1.
--------------------------Level 3	-------------------------------
	Display MS4-2 menu "SEC2". Three options: - Compute the slick velocity - Read the slick velocity, waveheight and windspeed - Exit

Users action	Computers action
! Select "Compute the slick velocity"	
	Call subroutine VELOX. Read the following files: DW1:(DF)WINDS.DAT DW1:(DF)WAVES.DAT DW1:(DF)TIDES.DAT DW1:(DF)CURRE.DAT Compute the resulting slick velocity and write result to the following file: DW1:(DF)VELOX.DAT Return to level 3.
→ Select "Read the slick velocity, waveheight and windspeed"	
	Read the X and Y coordinates of the slick velocity from the following file: DW1:(DF)VELOX.DAT Read windspeed array from the following file: DW1:(DF)WINDS.DAT Read the waveheight array from the following file: DW1:(DF)WAVES.DAT Go to level 4.
e Select "Exit"	
	Exit MS4-2. Return to level 1.

-------------------------Level 4-------------------------------

Display MS4-2 menu "SEC3".
Three options:
- List the slick transport file
- Commence simulation
- Exit

Users action	Computers action
* Select "List the slick transport file"	
	Call subroutine LIST3. Prints the following file: DW1:(DF)VELOX.DAT Return to level 4.
→ Select "Commence simulation"	
	Simulation takes place and all results are written to the following new files: DW1:(OUT)TRANSP.DAT DW1:(OUT)SPREAD.DAT DW1:(OUT)AGING1.DAT DW1:(OUT)AGING2.DAT DW1:(OUT)AGING3.DAT All the internal used parameters are written to the file: DW1:(OUT)PARAME.DAT Return to level 1.
e Select "Exit"	
	Exit MS4-2. Return to level 1.

B.4. MS4-3: Output presentation

Users action	Computers action
Switch on computer and lineprinter	
	Display main menu P/OS V1.5 Eight options: - Prose - Basic - ..... - Extra applications - .....
! Select "Extra applications"$^{\lambda}$	
	Display menu with the application groups. Eleven options: - A Group - B Group - C Group - ....
! Select "B Group"	

------------------------Level 1--------------------------------

	Display menu with the MS4 applications. Three options: - MS4-1: Data accumulation and processing - MS4-2: Actual simulation - MS4-3: Output presentation
! Select "MS4-3: Output presentation"	

ad λ In the present-day simulation the directory MS6 has to be selected prior to this step. This condition - to select the right directory - will be eliminated in the near future.

Users action	Computers action
-----------------------Level 2	-------------------------------
	Display MS4-3 menu "MENU1". Five options: - Listing transport file - Listing spreading file - Listing aging files - Listing parameter file - Exit
* Select "Listing transport file"	
	Print the transport file: DW1:(OUT)TRANSP.DAT Return to level 2.
* Select "Listing spreading file"	
	Print the spreading file: DW1:(OUT)SPREAD.DAT Return to level 2.
* Select "Listing aging files"	
	Print the aging files. DW1:(OUT)AGING1.DAT DW1:(OUT)AGING2.DAT DW1:(OUT)AGING3.DAT Return to level 2.
* Select "Listing parameter file"	
	Print the parameter file. DW1:(OUT)PARAME.DAT Return to level 2.
e Select "Exit"	
	Exit MS4-3. Return to level 1.

B.5. Alternative run procedure

The alternative procedure to run a simulation module (MS4-1, MS4-2 or MS4-3) is by selecting the tool kit from the main menu and after installation of the tool kit giving the following commands:

```
$ set def (MS4)                 ; specify directory of your choice
                                  for instance MS4
$ install (zzsys)prof77.tsk

$ install (zzsys)cglfpu.tsk

$ install MS4.tsk

$ run MS4.tsk
```

Appendix C: List of all files plus program file structure

C.1. Introduction

Before listing all MS4 files the process of transforming a FORTRAN 77 program into an executing task will be illustrated.

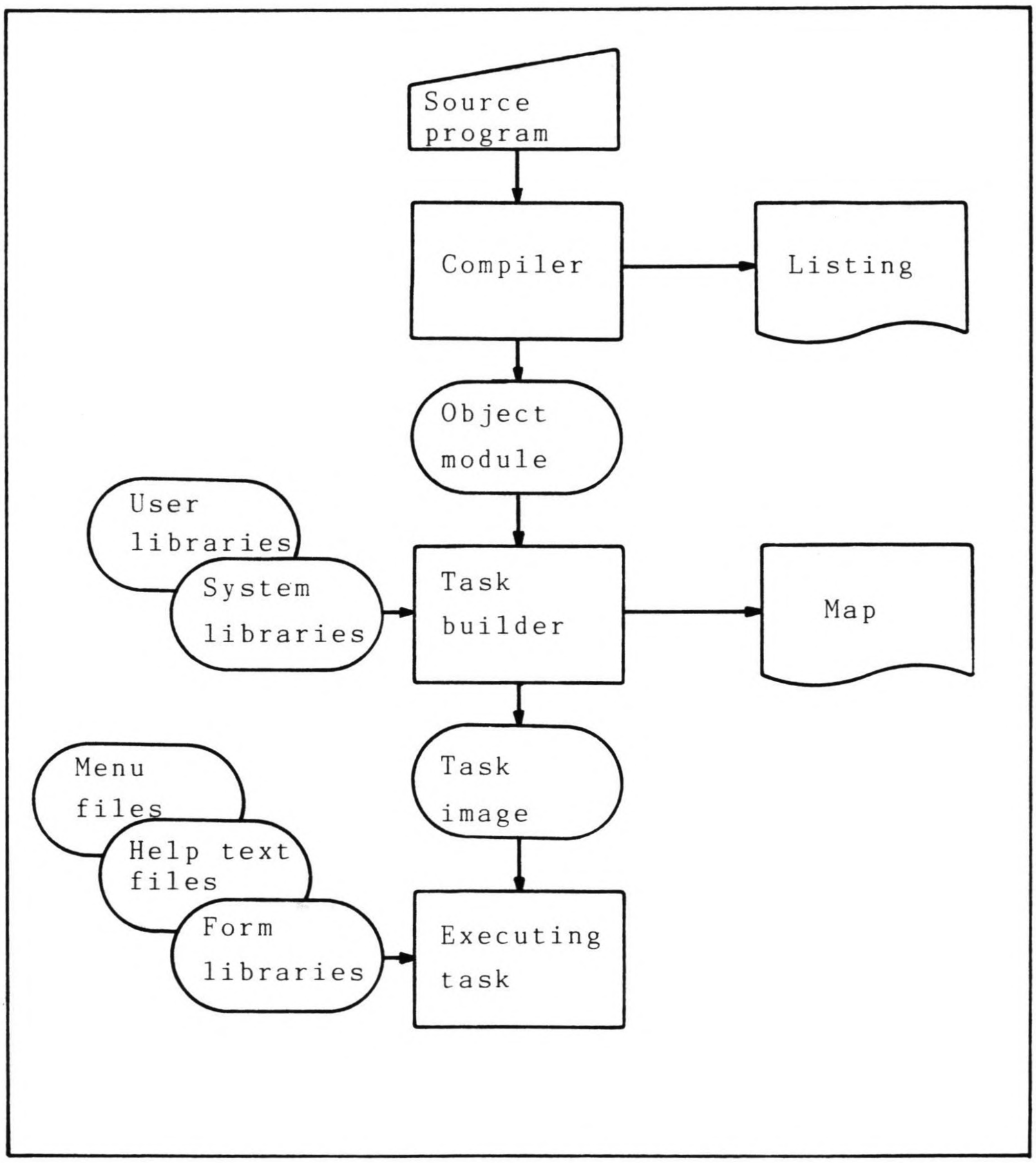


Figure C.1. Preparing a FORTRAN 77 program for execution

The format of a filespecification is as follows:

Device:(Directory)Filename.Filetype;Version

Device : The device on which a file is stored or is to be written. In the case of MS4 this is always the winchester disk, i.e. DW1:.

Directory: The directory by its name.
MS4 - refers to MS4-1
MS5 - refers to MS4-2
MS6 - refers to MS4-3

Filename : The file by its name.
DAP - refers to a data accumulation program
INT - refers to an interpolation program
LIST - refers to a listing program

Filetype : The kind of data in the file.
FTN - refers to a FORTRAN 77 source program
OBJ - refers to the object module
TSK - refers to the task image
MNQ - refers to an original menu file
MNU - refers to a converted menu file
HLQ - refers to an original help text file
HLP - refers to a converted help text file
FLB - refers to a form library
CMD - refers to the command file
INS - refers to the installation file
ODL - refers to the overlay descriptor file
DAT - refers to a data file

Version : The version of the file that is desired. Versions are identified by an octal number, which is incremented by 1 each time a new version of a file is created. The version number is important for distinguishing seperate runs.

C.2. List of all files related to MS4-1

Filename	Filetype	
MS4	.FTN	- Source programs
DAPTIM	.FTN	
DAPGEO	.FTN	
DAPSLK	.FTN	
DAPOIL	.FTN	
DAPSEA	.FTN	
DAPWEA	.FTN	
DAPWND	.FTN	
DAPWAV	.FTN	
DAPTID	.FTN	
DAPCUR	.FTN	
DAPCOM	.FTN	
INTWND	.FTN	
INTWAV	.FTN	
INTTID	.FTN	
INTCUR	.FTN	
LIST1	.FTN	
LIST2	.FTN	
MS4	.CMD	- System files
MS4	.INS	
MS4	.ODL	
MS4	.MNQ	- Menu files
MS4	.MNU	
MS4	.HLQ	- Help text files
MS4	.HLP	
FRMTIM	.FLB	- Form libraries
FRMGEO	.FLB	
FRMSLK	.FLB	
FRMOIL	.FLB	
FRMSEA	.FLB	
FRMWEA	.FLB	

FRMWND	.FLB	- Form libraries
FRMWAV	.FLB	
FRMTID	.FLB	
FRMCUR	.FLB	
FRMCOM	.FLB	
MS4	.OBJ	- Object modules
DAPTIM	.OBJ	
DAPGEO	.OBJ	
DAPSLK	.OBJ	
DAPOIL	.OBJ	
DAPSEA	.OBJ	
DAPWEA	.OBJ	
DAPWND	.OBJ	
DAPWAV	.OBJ	
DAPTID	.OBJ	
DAPCUR	.OBJ	
DAPCOM	.OBJ	
INTWND	.OBJ	
INTWAV	.OBJ	
INTTID	.OBJ	
INTCUR	.OBJ	
LIST1	.OBJ	
LIST2	.OBJ	
MS4	.TSK	- Task

C.3. List of all files related to MS4-2

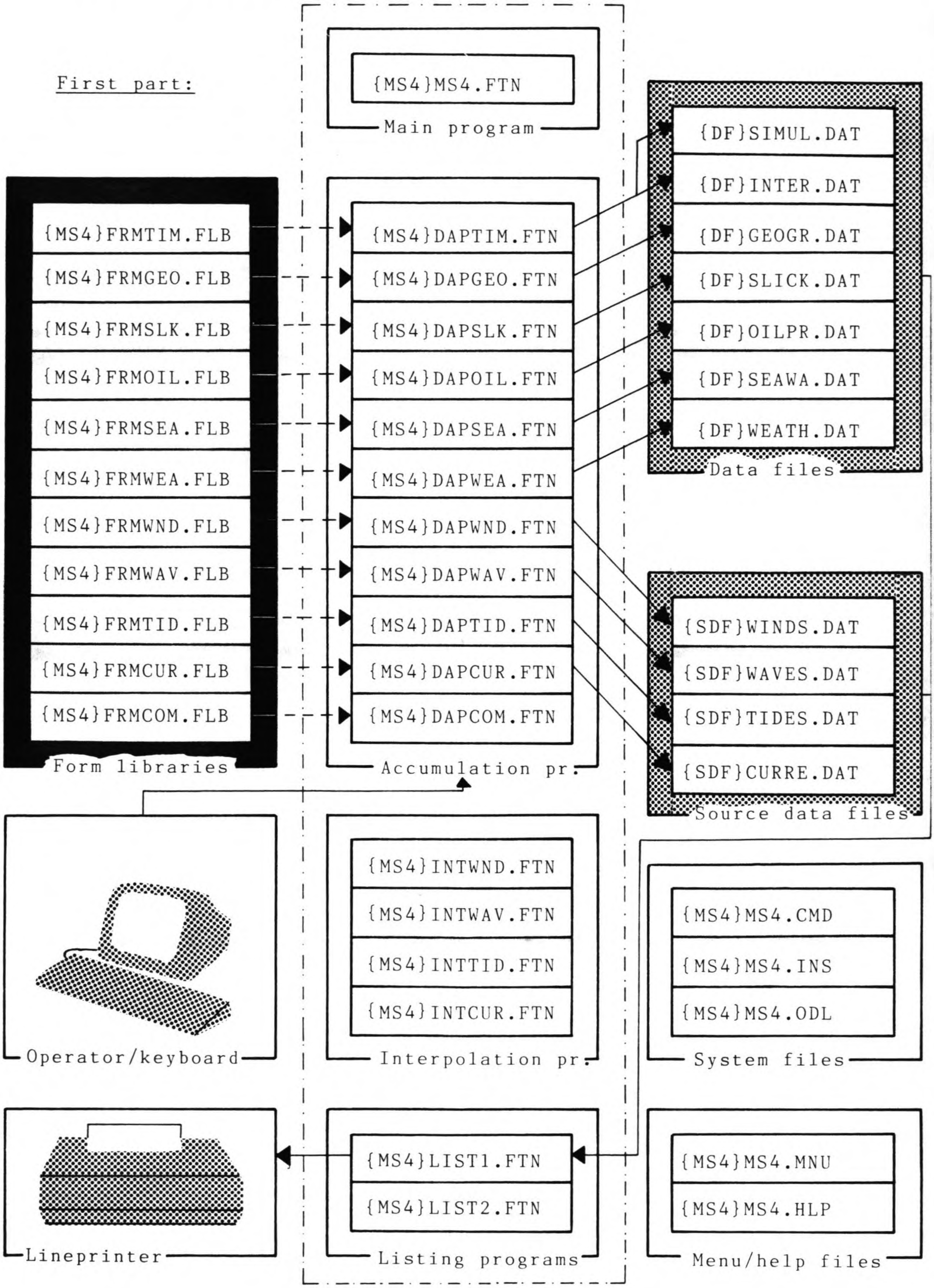
Filename	Filetype	
MS5	.FTN	- Source programs
VELOX	.FTN	
LIST3	.FTN	
MS5	.CMD	- System files
MS5	.INS	
MS5	.ODL	
MS5	.MNQ	- Menu files
MS5	.MNU	
MS5	.HLQ	- Help text files
MS5	.HLP	
MS5	.OBJ	- Object modules
VELOX	.OBJ	
LIST3	.OBJ	
MS5	.TSK	- Task

C.4. List of all files related to MS4-3

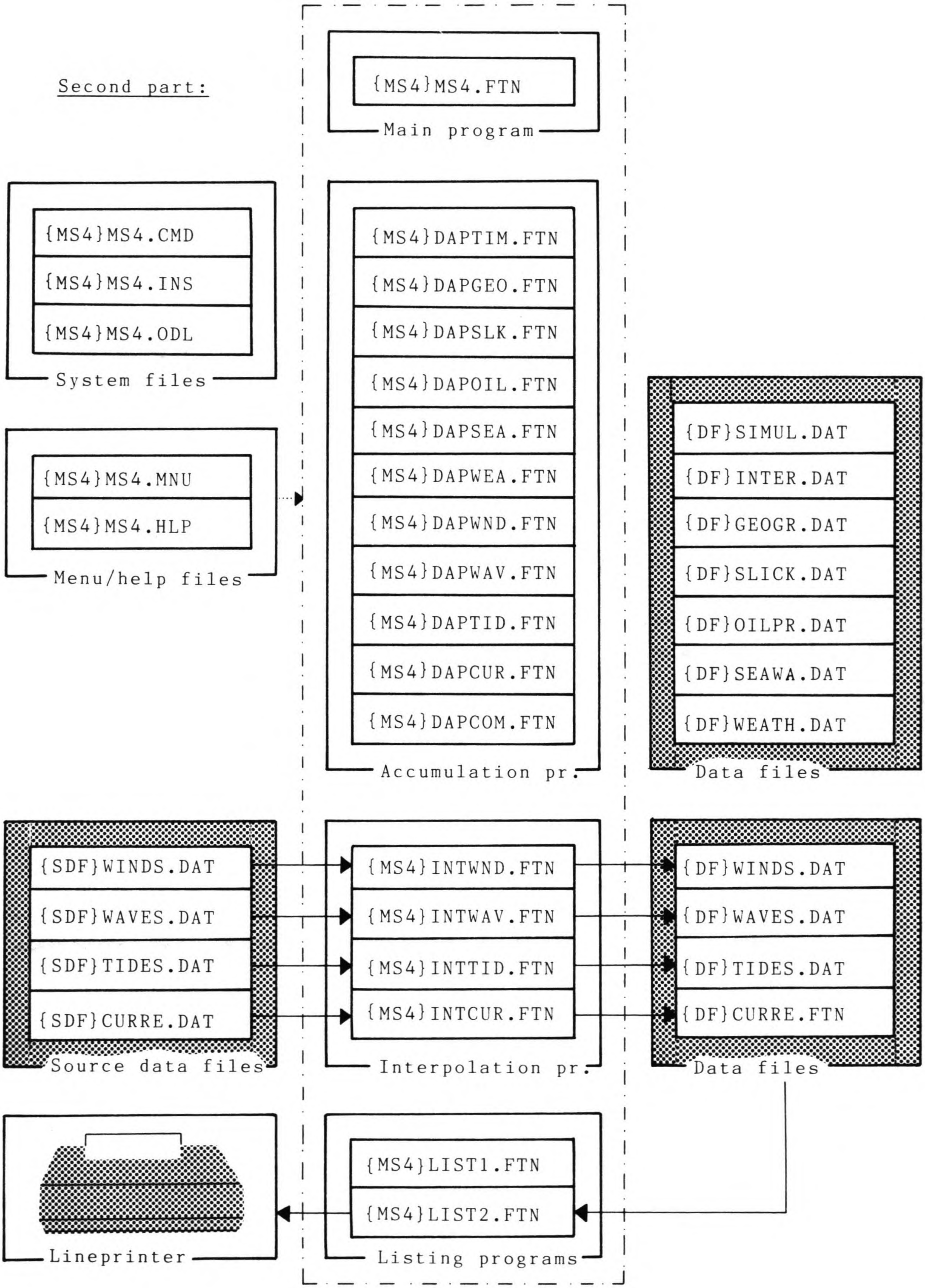
Filename	Filetype	
MS6	.FTN	- Source program
MS6	.CMD	- System files
MS6	.INS	
MS6	.ODL	
MS6	.MNQ	- Menu files
MS6	.MNU	
MS6	.HLQ	- Help text files
MS6	.HLP	
MS6	.OBJ	- Object module
MS6	.TSK	- Task

C.5. Program file structure: MS4-1

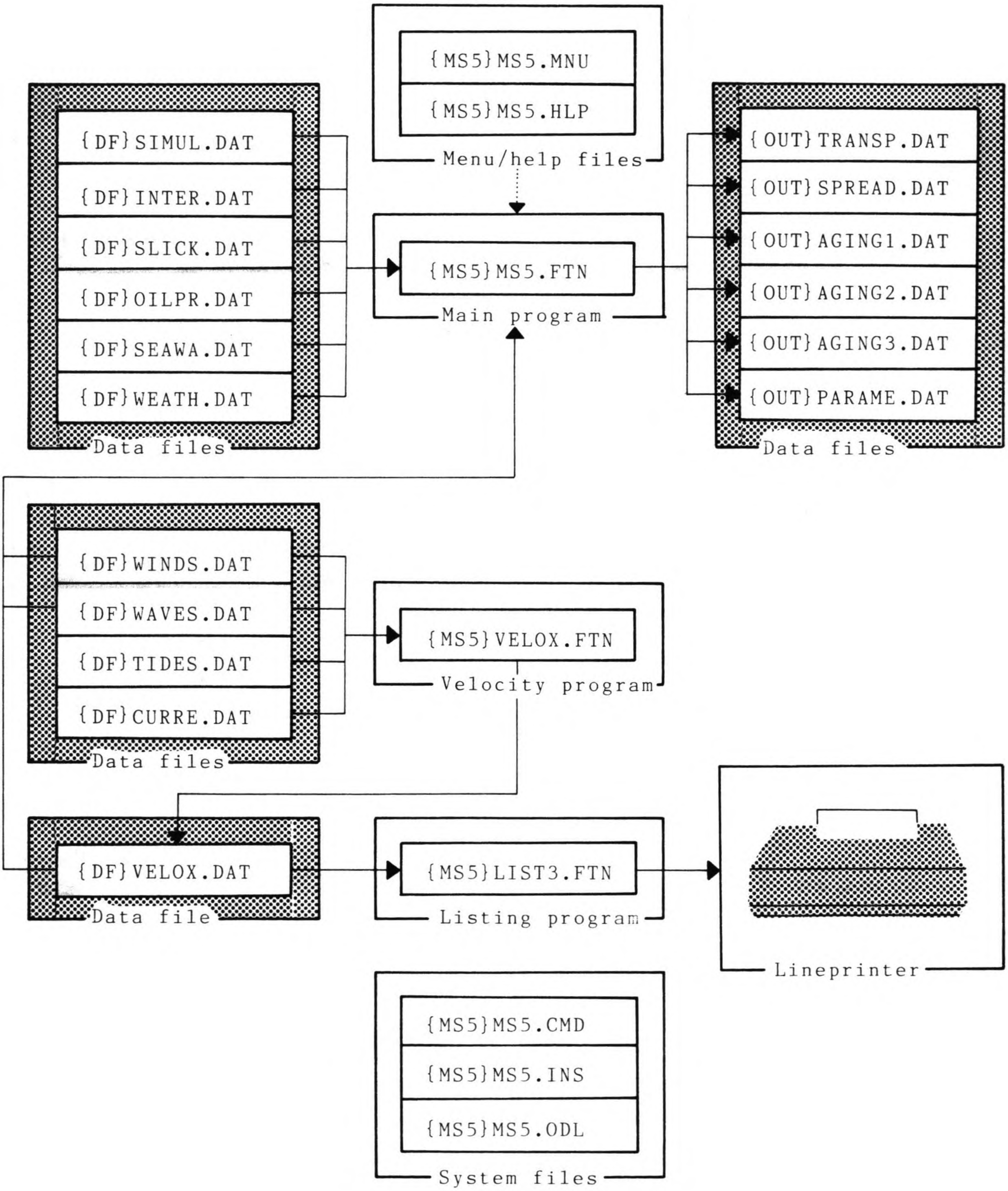
First part:



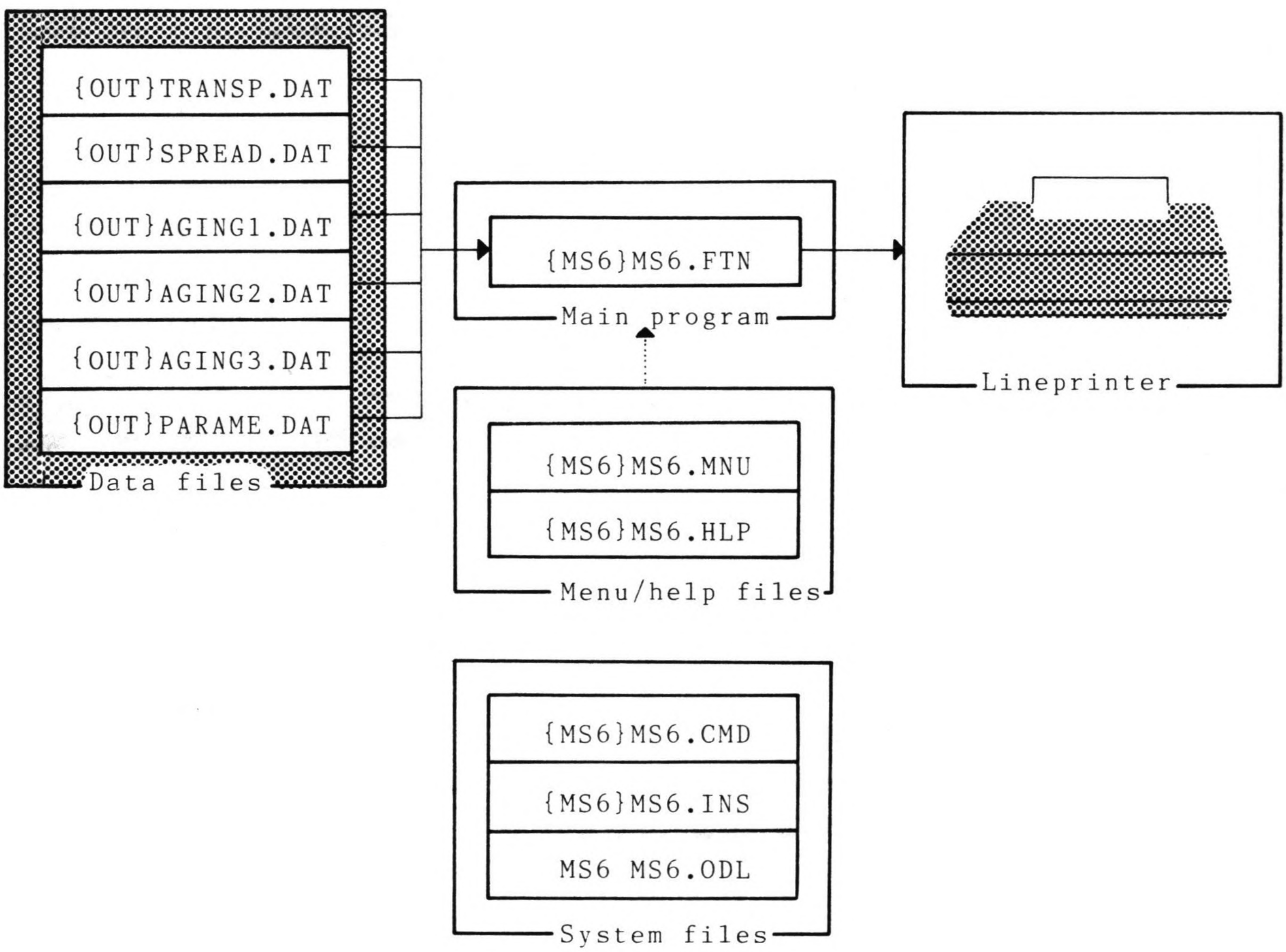
Second part:


{MS4}MS4.FTN
Main program
{MS4}MS4.CMD
{MS4}MS4.INS
{MS4}MS4.ODL
System files
{MS4}MS4.MNU
{MS4}MS4.HLP
Menu/help files
{MS4}DAPTIM.FTN
{MS4}DAPGEO.FTN
{MS4}DAPSLK.FTN
{MS4}DAPOIL.FTN
{MS4}DAPSEA.FTN
{MS4}DAPWEA.FTN
{MS4}DAPWND.FTN
{MS4}DAPWAV.FTN
{MS4}DAPTID.FTN
{MS4}DAPCUR.FTN
{MS4}DAPCOM.FTN
Accumulation pr.
{DF}SIMUL.DAT
{DF}INTER.DAT
{DF}GEOGR.DAT
{DF}SLICK.DAT
{DF}OILPR.DAT
{DF}SEAWA.DAT
{DF}WEATH.DAT
Data files
{SDF}WINDS.DAT
{SDF}WAVES.DAT
{SDF}TIDES.DAT
{SDF}CURRE.DAT
Source data files
{MS4}INTWND.FTN
{MS4}INTWAV.FTN
{MS4}INTTID.FTN
{MS4}INTCUR.FTN
Interpolation pr.
{DF}WINDS.DAT
{DF}WAVES.DAT
{DF}TIDES.DAT
{DF}CURRE.FTN
Data files
Lineprinter
{MS4}LIST1.FTN
{MS4}LIST2.FTN
Listing programs

C.6. Program file structure: MS4-2



C.7. <u>Program file structure: MS4-3</u>



Appendix D: Technical specifications Professional 350

Manufacturer	Digital Equipment Corporation
System	Professional 350
Processor	F-11
Memory	512 Kb
Floppy disks	$5\frac{1}{4}$ inch
	2 x 400 Kb
Winchester disk	10 Mb standard
Floating point processor	standard
Printer gate	RS232-C
Operating system	P/OS
Video monitor	12 inch black/white
Bit map graphics	standard